

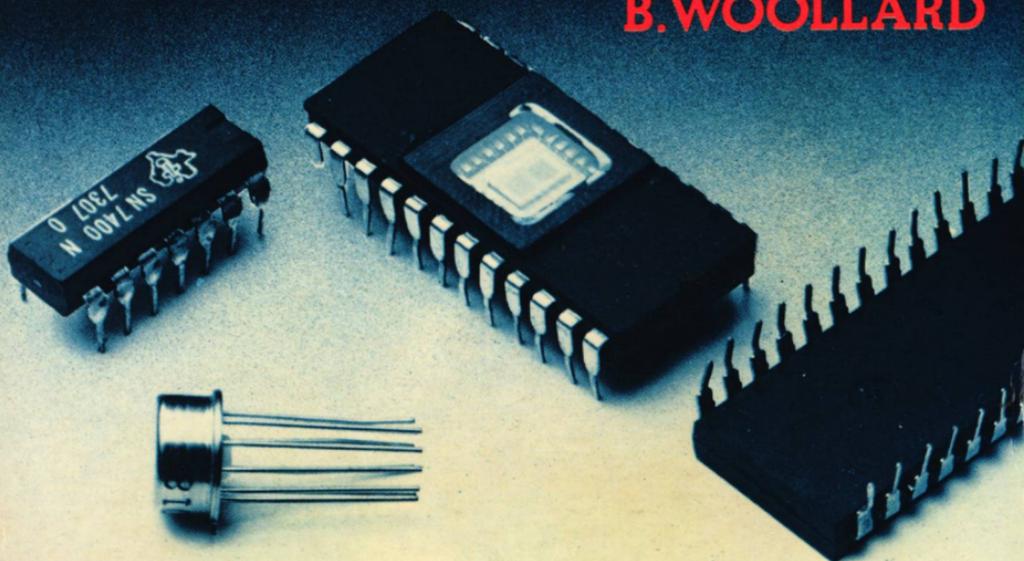
dunod

PRATIGUIDE

électronique

Conquérir la logique

B. WOOLLARD



Du circuit intégré à l'ordinateur

Conquérir la logique

Collection

dunod
PRATIGUIDE
électronique

Apprivoiser les composants, B. Woollard

Les connaître pour les utiliser

Auto montages, G. Bishop

Equiper, améliorer sa voiture

Calculer ses circuits, R.G. Krieger

Des formules sans problème

Conquérir la logique, B. Woollard

Des circuits intégrés à l'ordinateur

Construire ses premiers kits, Y. Dang et J.-C. Fantou

Des gadgets pour s'initier à l'électronique

Randonnée électronique, J. Watson

Des montages pour les loisirs

Réussir ses circuits imprimés, J. Goldberg

Et simplifier ses montages

Sonoriser par le kit, Y. Dang et J.-C. Fantou

De l'amplificateur à l'égaliseur

Tester et mesurer, T. Dixon

Comment réaliser son équipement

B.WOOLLARD

Conquérir la logique

Du circuit intégré à l'ordinateur

Traduit par J.C.Fantou

dunod
PRATIGUIDE
electronique

Traduction française de l'ouvrage publié
en langue anglaise sous le titre :
DIGITAL INTEGRATED CIRCUITS AND COMPUTERS
par
MacGraw Hill Book Company (UK) Limited.

© **McGRAW HILL BOOK COMPANY (UK), Ltd, 1978.**
© **BORDAS, Paris, 1980 pour l'édition française et ses annexes.**
ISBN 2-04-010 665-0

" Toute représentation ou reproduction, intégrale ou partielle, faite sans le consentement de l'auteur, ou de ses ayants-droit, ou ayants-cause, est illicite (loi du 11 mars 1957, alinéa 1^{er} de l'article 40). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait une contrefaçon sanctionnée par les articles 425 et suivants du Code pénal. La loi du 11 mars 1957 n'autorise, aux termes des alinéas 2 et 3 de l'article 41, que les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective d'une part, et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration "

en bref

La révolution micro-informatique qui se produit aujourd'hui met la technologie au premier rang dans tous les domaines de l'industrie — quelle que soit leur vocation de base — et même au sein des familles où l'ordinateur individuel vient supplanter rapidement le train électrique dans les loisirs.

Si cet envahissement de la vie quotidienne est assez bien accepté, il est moins facile d'aborder directement les systèmes micro-informatiques proposés, surtout sur le plan du matériel, sans connaissances solides dans le domaine de la logique. Traiter sereinement les problèmes d'adaptations ou de maintenance passe par l'apprentissage de l'emploi d'un bon nombre d'éléments standards qui existent à l'état intégré et dont il est utile d'analyser le fonctionnement.

Cet ouvrage veut donner les bases fondamentales qui permettront de maîtriser microprocesseurs et micro-ordinateurs.

découverte

des thèmes essentiels

1. des puces au silicium	1
• <i>Technologie des circuits</i>	1
Développement des circuits intégrés.....	2
Encapsulation des circuits intégrés.....	11
Abréviations d'usage courant.....	13
2. de porte en porte	14
• <i>Portes logiques de base</i>	14
L'algèbre de Boole.....	15
Les règles de base de l'algèbre de Boole.....	15
Les lois de la logique.....	16
La représentation symbolique des éléments de logique.....	16
Portes de base et tables de vérité.....	18
Systèmes de numération.....	21
3. des problèmes de famille	24
• <i>Circuits intégrés numériques</i>	24
Choix d'une famille logique.....	24
Vitesse de fonctionnement.....	25
L'immunité au bruit.....	27
L'entrance et la sortance.....	28
La consommation.....	29
Les familles logiques.....	29
La logique à diodes et résistances (DRL).....	30

La logique à résistances et transistors (RTL).....	34
La logique à diodes et transistors (DTL).....	35
La logique tout à transistors (TTL).....	38
Courants d'évacuation et de source.....	40
La désignation des circuits TTL.....	42
Feuilles de caractéristiques d'un circuit TTL.....	43
Logique non saturée à couplage par les émetteurs (ECL).....	44
Logique à transistors MOS.....	45
Logique à MOS complémentaires (CMOS).....	46
4. toute la vérité.....	49
• <i>Construire les logigrammes.....</i>	<i>49</i>
Portes ET-NON (NAND) et OU-NON (NOR).....	49
Porte OU EXCLUSIF.....	53
• <i>Les tableaux de Karnaugh.....</i>	<i>59</i>
Représentation graphique des fonctions.....	61
Regroupement des cases.....	61
5. les bascules en balance.....	73
• <i>Différents types de bistables.....</i>	<i>73</i>
Bascules RS.....	73
Bascules RST.....	76
Bascules D.....	76
Bascules JK.....	78
6. une lourde addition.....	81
• <i>L'arithmétique binaire.....</i>	<i>81</i>
L'addition binaire.....	81
L'additionneur série.....	84
L'additionneur parallèle.....	86
La soustraction binaire.....	87
Le soustracteur binaire.....	89
La multiplication binaire.....	91
La division binaire.....	91
Le multiplicateur d'impulsions.....	94
7. des registres sur qui compter.....	95
• <i>Les registres et les compteurs.....</i>	<i>95</i>
Le registre-mémoire.....	95
Le registre à décalage.....	98
Commande d'écriture d'un registre à décalage.....	100
Registre à décalage réversible.....	104
Le compteur asynchrone.....	108
Le compteur-décompteur.....	110
Le décodage.....	110

Le compteur synchrone.....	112
Code « décimal codé binaire » (BCD).....	117
Compteur en BCD.....	118
Compteur en anneau et compteur Johnson.....	119
Générateurs pseudo aléatoires.....	127
8. histoires de logique.....	135
• <i>Quelques applications.....</i>	<i>135</i>
Un système de sécurité.....	137
Contrôle de température.....	137
Sonnette de service.....	138
Simulation d'une commande d'ascenseur.....	141
Passage à niveau à commande automatique.....	145
Passage protégé (pour piétons).....	148
9. des bêtes de somme.....	151
• <i>Généralités sur les ordinateurs.....</i>	<i>151</i>
Qu'est-ce qu'un ordinateur?.....	151
Comment fonctionne un ordinateur?.....	152
Exactitude et précision des informations d'entrée.....	152
Éléments d'un ordinateur.....	153
L'unité centrale de traitement.....	154
Les instructions.....	155
Les mémoires.....	155
Unité arithmétique et logique (ALU).....	156
L'unité de commande.....	156
Les unités périphériques d'entrée.....	157
Les unités périphériques de sortie.....	158
Applications.....	158
Le développement des ordinateurs.....	159
• <i>Le tour des mémoires.....</i>	<i>160</i>
Les mémoires à tores de ferrite.....	161
Lecture et écriture d'une mémoire.....	163
Mémoire à film mince.....	165
Mémoire à semi-conducteurs.....	165
Mémoire RAM statique.....	166
Mémoire RAM dynamique.....	168
Mémoires ROM, PROM, EPROM et EAROM.....	171
Mémoires auxiliaires.....	174
Mémoire à tambour magnétique.....	175
Mémoire à bande magnétique.....	177
Mémoire à disques magnétiques.....	178
• <i>Concevoir sa programmation.....</i>	<i>180</i>
L'organigramme.....	181
Symboles arithmétiques.....	182
Expressions algébriques.....	184

Bouclage.....	184
Le programme.....	188
Conception d'un logiciel.....	189
Préparation d'un programme.....	194
10. et pour conclure.....	195
• <i>Quelques compléments</i>	195
Symboles logiques.....	195
Brochage des circuits de la série TTL 74.....	195
Quelques derniers conseils.....	196
Générateur d'horloge 555 et circuits divers.....	198
Index.....	205

des puces au silicium

technologie des circuits

Les circuits intégrés (CI) se divisent en deux grandes catégories :

- les circuits intégrés *linéaires* comprenant des circuits électroniques d'amplification,
- les circuits intégrés numériques à base de commutation.

Cet ouvrage ne traite que des circuits intégrés numériques, c'est-à-dire des circuits capables de traiter des informations numériques par l'intermédiaire de circuits électroniques de commutation. Il existe toutefois un domaine dans lequel les circuits numériques n'ont pas encore bien pénétré : c'est celui de la commande de puissance des machines industrielles (les moteurs, par exemple), car la puissance qu'ils peuvent traiter est encore limitée.

Les circuits intégrés numériques servent à traiter et à mémoriser l'information dans les systèmes numériques (ordinateurs, calculatrices de bureau ou de poche, commandes numériques de machines-outils, appareils de mesure numérique, etc. L'utilisation de circuits intégrés accélère et facilite la conception d'un ensemble, car le concepteur n'a plus à assembler des composants actifs et passifs pour réaliser portes et bascules. Ces fonctions existent déjà toutes faites. Le concepteur ne fait pas non plus l'assemblage de portes et de bascules pour réaliser un ensemble plus complexe; des circuits intégrés existent qui réalisent directement des fonctions de base comprenant déjà des portes et des bascules sur une « puce » semi-conductrice. Son travail consiste à assembler des fonctions de base disponibles sous forme de circuits intégrés.

Ainsi l'étude des circuits intégrés numériques passe obligatoirement par l'apprentissage des fonctions de base les plus courantes.

Le développement des circuits intégrés

Le développement des circuits intégrés a suivi l'évolution des technologies de fabrication des transistors. Les premiers types de tran-

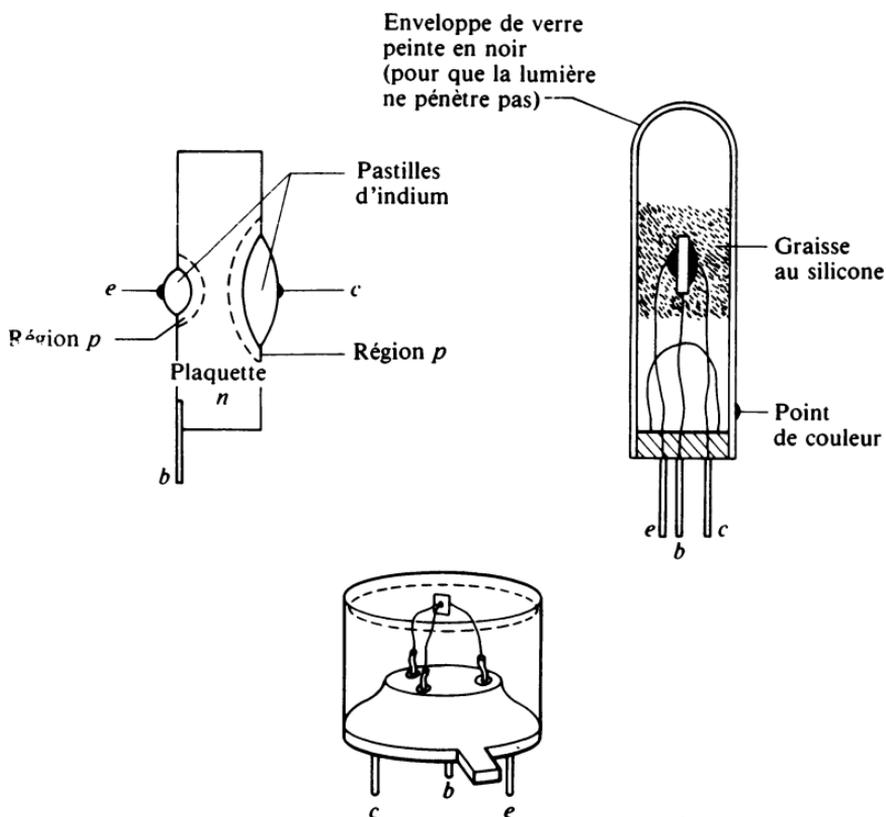


Fig. 1 Transistor à alliage

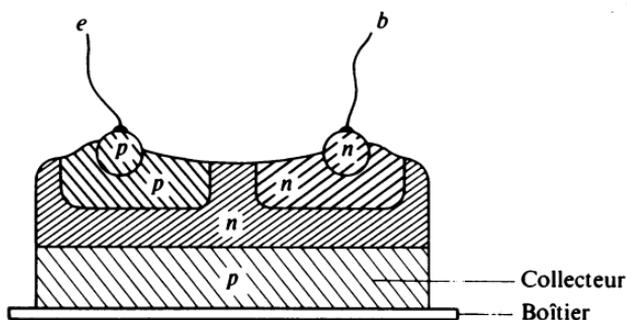


Fig. 2 Transistor à alliage et diffusion.

sistors étaient à croissance, c'est-à-dire que les régions p et n étaient réalisées par croissance à partir d'un même cristal de semi-conducteur. Puis, vinrent les transistors à alliage, dans lesquels deux pastilles d'impuretés de type p étaient alliées sur les faces opposées d'une plaquette de semi-conducteur de type n , donnant une épaisseur de base d'environ $25\ \mu\text{m}$, tel que l'indique la figure 1. Cette technologie améliorait les performances haute fréquence, sans toutefois dépasser le seuil de 1 MHz. Les transistors à alliage et diffusion devaient permettre de repousser ce seuil jusqu'à plusieurs centaines de mégahertz, car ils avaient une épaisseur de base plus faible, tel que le montre la figure 2. Toutefois, ces transistors supportaient des tensions collecteur-émetteur assez faibles. Le procédé *planar*, qui est à l'heure actuelle le seul procédé utilisé pour la fabrication des transistors au silicium, fut décrit pour la première fois en 1961. Il permet de réaliser des transistors ayant d'excellentes performances haute fréquence — dépassant le gigahertz — associées à une très bonne tenue en tension, puisque certains transistors spéciaux supportent des tensions collecteur-émetteur de plusieurs milliers de volts. Cette technologie de fabrication présente également l'avantage de permettre de réaliser

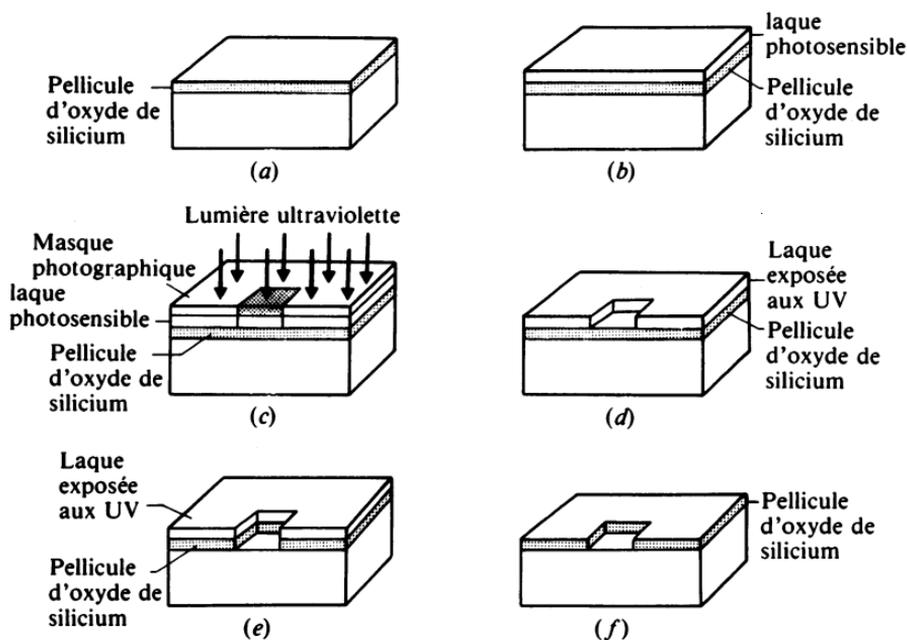


Fig. 3 Procédé de photogravure :

- (a) tranche de silicium oxydée,
- (b) application d'une laque photosensible,
- (c) exposition de la laque aux UV à travers un masque photographique,
- (d) enlèvement de la laque photosensible au solvant,
- (e) gravure de la pellicule d'oxyde de la fenêtre,
- (f) enlèvement de la laque.

simultanément plusieurs milliers de transistors sur une même plaque de semi-conducteur.

Le procédé de fabrication d'un transistor planar est le suivant : une pellicule d'oxyde est déposée sur la surface d'une tranche de semi-conducteur de type n . Des fenêtres sont ensuite gravées dans la pellicule d'oxyde par un procédé de photogravure, tel que l'indique la figure 3.

La tranche de semi-conducteur est ensuite exposée à une atmosphère de type p , et la région de base est diffusée à travers la fenêtre

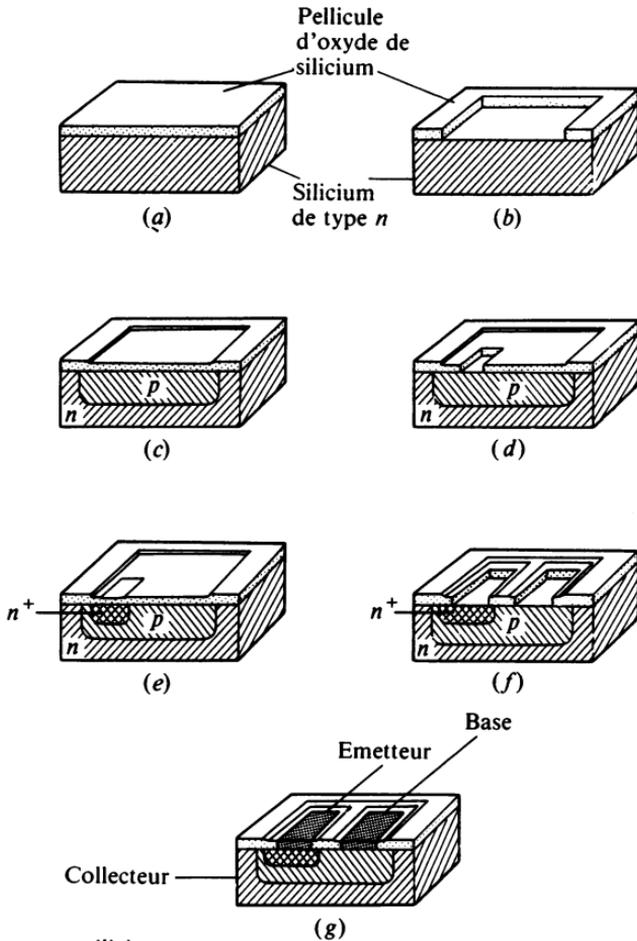


Fig. 4 Procédé planar au silicium :

- (a) Silicium de type n oxydé,
- (b) découpe de la fenêtre de base lors de la première étape de gravure,
- (c) diffusion de la région de base de type p et formation d'une nouvelle pellicule d'oxyde,
- (d) découpe de la fenêtre d'émetteur lors de la deuxième étape de gravure,
- (e) diffusion de l'émetteur n^+ et formation d'une nouvelle pellicule d'oxyde,
- (f) découpe des fenêtres de contact de base et d'émetteur (troisième étape de gravure),
- (g) évaporation et définition des contacts en aluminium (quatrième étape de gravure).

dans la tranche de type n . La surface est oxydée de nouveau, la fenêtre d'émetteur est gravée et la région n^+ à faible résistivité de l'émetteur est diffusée dans la région de base (fig. 4). Les différents masques de réalisation d'un transistor planar, ainsi que leur ordre d'utilisation sont indiqués par la figure 5. Les bords des jonctions pn sont protégés par les différentes pellicules d'oxyde, ce qui évite la pollution des jonctions par l'environnement.

De façon à obtenir une faible capacité de collecteur, associée à une tension de claquage élevée, il est nécessaire que la région de collecteur présente une forte résistivité. Toutefois, une forte résistivité entraîne une valeur trop élevée de la résistance entre la jonction de collecteur et le contact de collecteur. Ce problème est résolu grâce au procédé épitaxial dans lequel une fine couche de semi-conducteur n à forte résistivité est déposée sur un substrat épais de semi-conducteur n^+ de faible résistivité. Le procédé de diffusion permet d'obtenir d'excellents résultats lorsqu'il s'agit de déposer une fine couche à faible résistivité sur des substrats à forte résistivité, mais il est mauvais dans l'autre sens. Le procédé épitaxial consiste à déposer en phase vapeur, du semi-conducteur à très haute pureté sur un substrat à moins grande pureté, tout en conservant la structure cristalline du substrat. Le procédé de fabrication d'un transistor commence donc par un substrat épais n^+ à faible résistivité sur lequel on dépose par épitaxie une couche n à forte résistivité. C'est dans cette couche qu'on va réaliser le transistor en employant le procédé planar précédent. On peut ne diffuser que dans la couche épitaxiale, réalisant alors les impératifs de

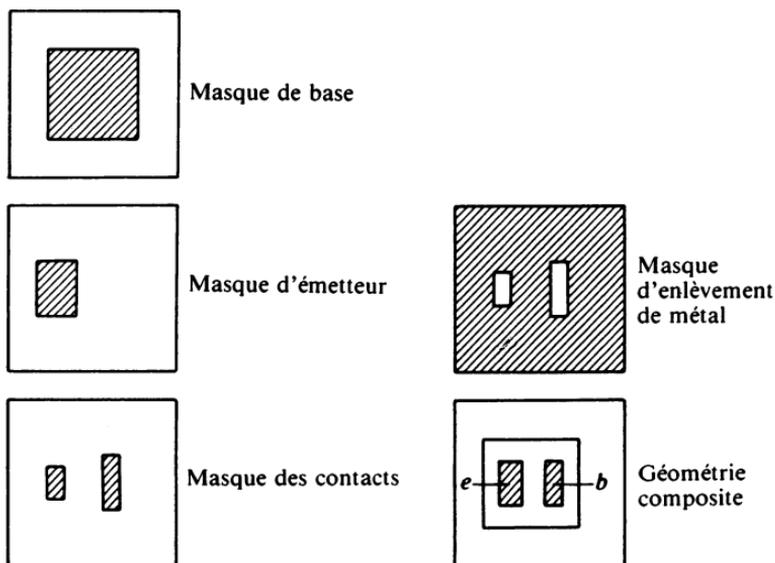


Fig. 5 Masques photographiques de réalisation d'un transistor planar au silicium.

faible capacité et forte tension de claquage. On obtient ainsi un transistor planar-épitaxial au silicium dont le substrat épais est maintenant à faible résistivité, tel que l'indique la figure 6.

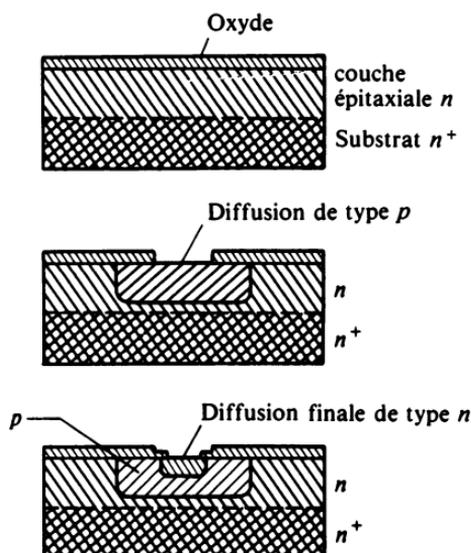


Fig. 6 Transistor planar-épitaxial au silicium.

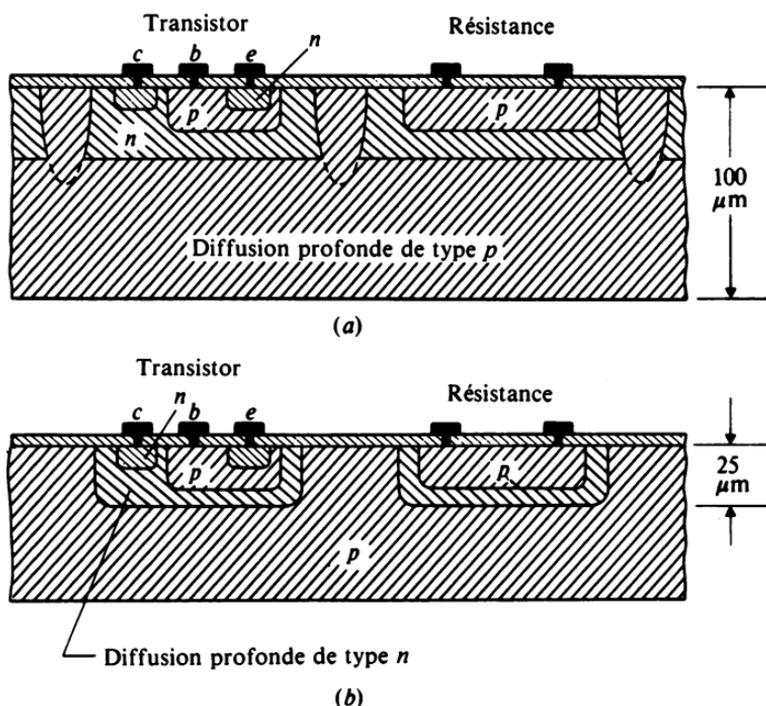


Fig. 7 Première génération de circuits intégrés :

- (a) procédé à diffusion profonde,
 (b) procédé à triple diffusion.

Ces méthodes permettent de réaliser des circuits complets, appelés circuits intégrés monolithiques, dans lesquels les résistances, les condensateurs et les diodes sont réalisés dans une même couche

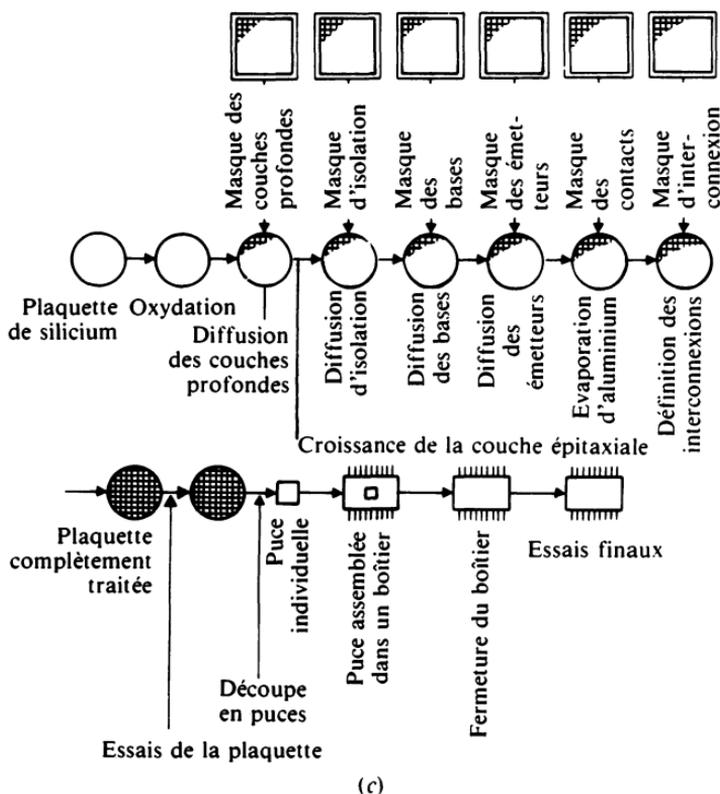
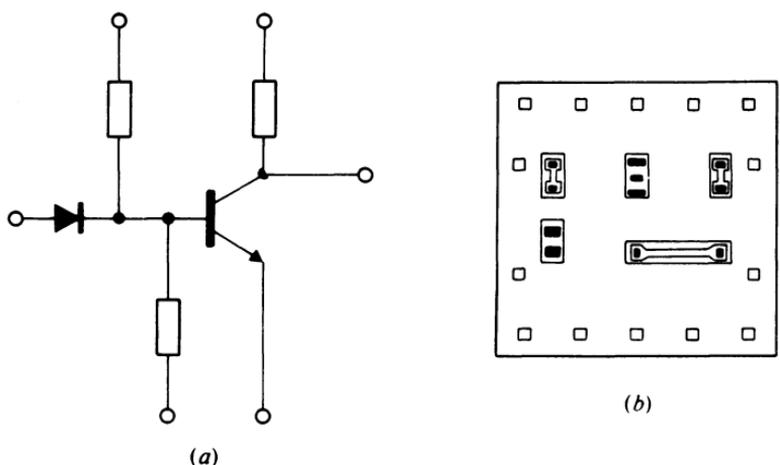


Fig. 8 Différentes phases de fabrication d'un circuit intégré :

- (a) conception du circuit,
- (b) disposition des éléments,
- (c) fabrication des masques.

épitaxiale et raccordés les uns aux autres par des connexions métalliques, déposées par évaporation sur la pellicule d'oxyde (fig. 7). La technologie de réalisation de ces circuits est dite *bipolaire*, car leur fabrication nécessite les deux types (*p* et *n*) de semi-conducteur. Les différentes phases de fabrication d'un circuit intégré sont illustrées par la figure 8 et un exemple de circuit intégré est donné en figure 9.

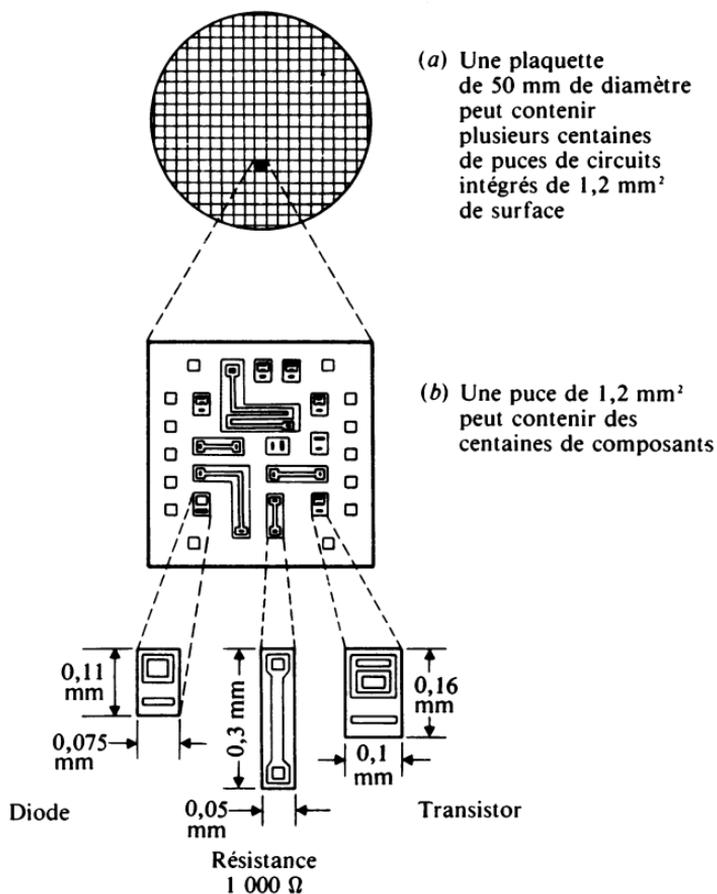


Fig. 9 Exemple de circuit intégré :

- (a) plaquette de circuits intégrés,
(b) puce de circuit intégré.

Plus récemment, des transistors *unipolaires* ont été développés. Ils sont dits unipolaires, car le courant qui les traverse est constitué uniquement de porteurs majoritaires. Le fonctionnement de ces transistors est basé sur l'action d'un champ électrique, et ils portent le nom de transistors à effet de champ (TEC en français, FET en anglais).

Un transistor à effet de champ à jonction, à canal *n*, comprend un barreau de semi-conducteur de type *n* (le canal), sur lequel on crée

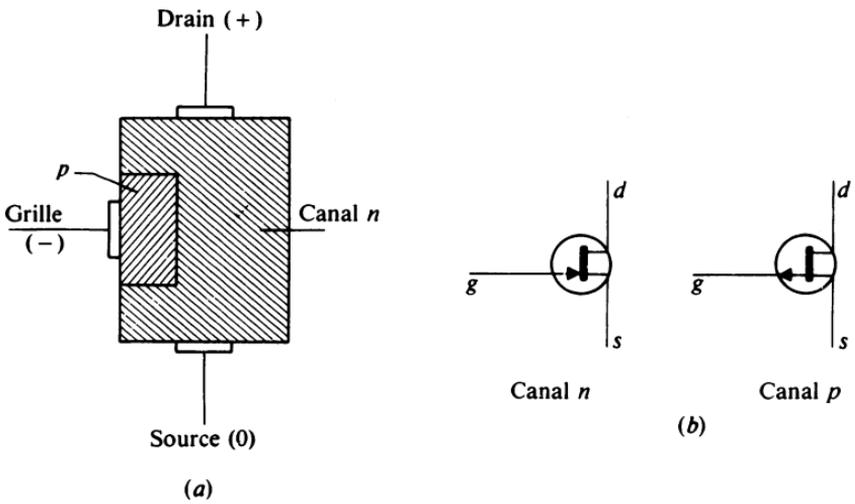


Fig. 10 Transistor à effet de champ à jonction (TEC) :

- (a) TEC à canal n ,
- (b) symbole d'un TEC

une région de type p (la grille), tel que le montre la figure 10. Les porteurs majoritaires circulent de la source (l'émetteur pour un transistor bipolaire) au drain (collecteur d'un transistor bipolaire). Une polarisation inverse appliquée à la jonction pn forme une zone à appauvrissement autour de la jonction, comme l'indique la figure 11 *a*. La zone à appauvrissement réduit la largeur effective du canal, ce qui revient à augmenter sa résistance, et donc à diminuer l'intensité du courant de drain. Toute augmenta-

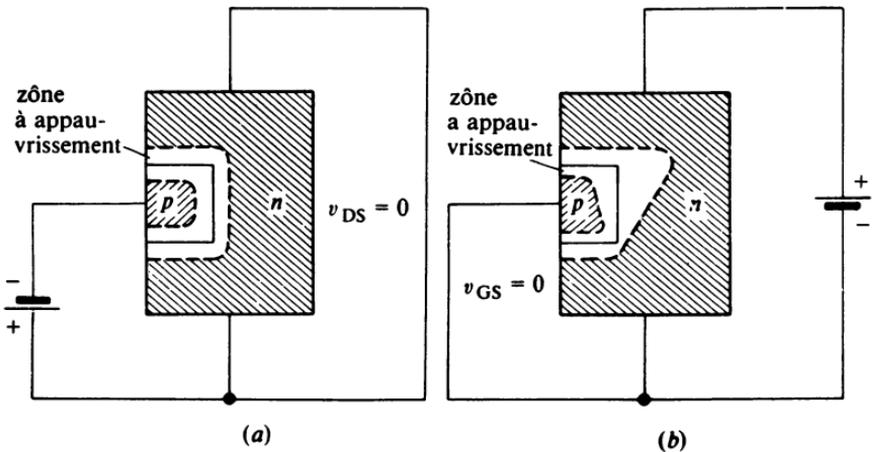


Fig. 11 Formation d'une zone à appauvrissement dans un TEC :

- (a) due à une tension inverse de grille,
- (b) due au champ électrique longitudinal.

tion de la tension de grille augmente la largeur de la zone à appauvrissement et de ce fait, réduit le courant de drain correspondant. Si on augmente la tension de drain, avec une tension de grille égale à zéro, on constate que le courant circulant longitudinalement dans le canal entraîne une différence de potentiel entre la grille et la portion du canal sur lequel se trouve la jonction de grille, différence de potentiel qui polarise en inverse la jonction et crée ainsi une zone à appauvrissement qui a l'allure indiquée par la figure 11 b. En augmentant cette fois la tension inverse de grille, on accroît la largeur de la zone à appauvrissement jusqu'à ce qu'elle « pince » le courant de drain. La tension de grille pour laquelle le courant de drain s'annule, porte le nom de *tension de pincement*. Pour les tensions de grille fortement positives, les caractéristiques de sortie indiquent la saturation. Ce fonctionnement correspond au mode à appauvrissement, puisque toute augmentation de la tension inverse de grille entraîne une augmentation de la zone à appauvrissement du canal. Un TEC peut fonctionner avec une très légère polarisation directe de grille, ce qui réduit la zone à appauvrissement et augmente la valeur du courant de drain. Ce mode de fonctionnement s'appelle le *mode à enrichissement*. L'avantage principal d'un TEC sur un transistor bipolaire réside dans la valeur élevée de l'impédance d'entrée (plusieurs centaines de mégohms), puisque le courant circulant dans la grille est nul par principe.

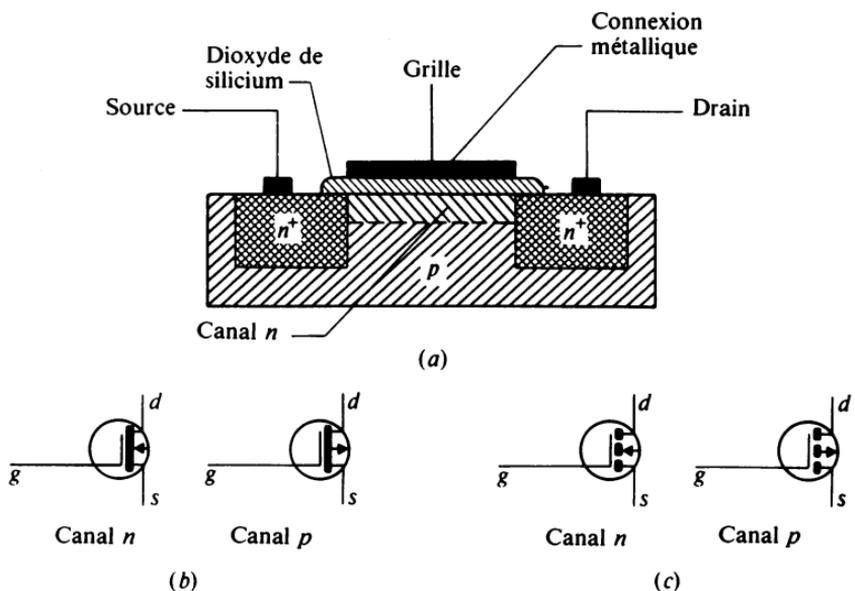


Fig. 12 Transistor à grille isolée (MOS) :
 (a) coupe d'un MOS,
 (b) symbole d'un MOS à appauvrissement,
 (c) symbole d'un MOS à enrichissement.

Un transistor à effet de champ à canal n à grille isolée est représenté sur la figure 12. La différence principale entre un transistor à grille isolée et un TEC réside dans le fait que la grille métallique est isolée du canal par une couche d'oxyde de silicium. C'est pour cette raison que l'on appelle ce type de transistor, un transistor métal-oxyde-silicium (MOS).

Un transistor MOS fonctionne le plus souvent en mode à enrichissement. Un potentiel positif appliqué sur la grille attire les porteurs minoritaires à surface de séparation du semi-conducteur et de l'oxyde, créant ainsi un canal d'inversion de type n dans le semi-conducteur de type p . Plus le potentiel de grille sera positif et plus la conductibilité du canal d'inversion sera grande. c'est-à-dire qu'une augmentation du potentiel de grille enrichit le courant circulant entre le drain et la source.

Les transistors MOS ont conduit au développement de circuits intégrés à MOS complémentaires (CMOS). On trouvera la description de ces transistors CMOS au chapitre 3.

Encapsulation des circuits intégrés

Trois types de boîtiers sont couramment utilisés pour les circuits intégrés :

- **boîtier TO5.** Ce boîtier est une version de hauteur réduite du boîtier TO5 très utilisé pour les transistors, disponible en version à 8 ou 10 broches (fig. 13).

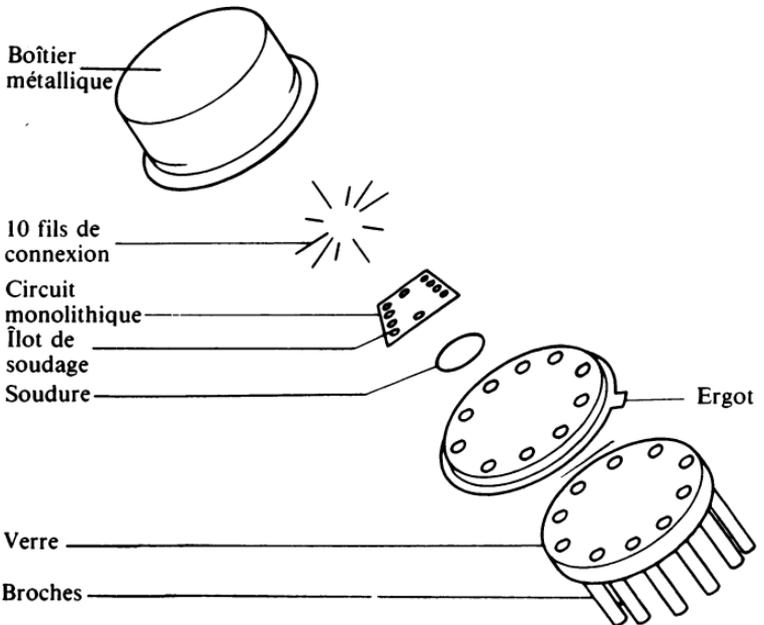


Fig. 13 Vue éclatée d'un boîtier TO5 à 10 broches.

• *boîtier plat* (14 broches). Ce boîtier a été conçu pour pouvoir être soudé sur les circuits imprimés, ce qui obligeait à respecter le

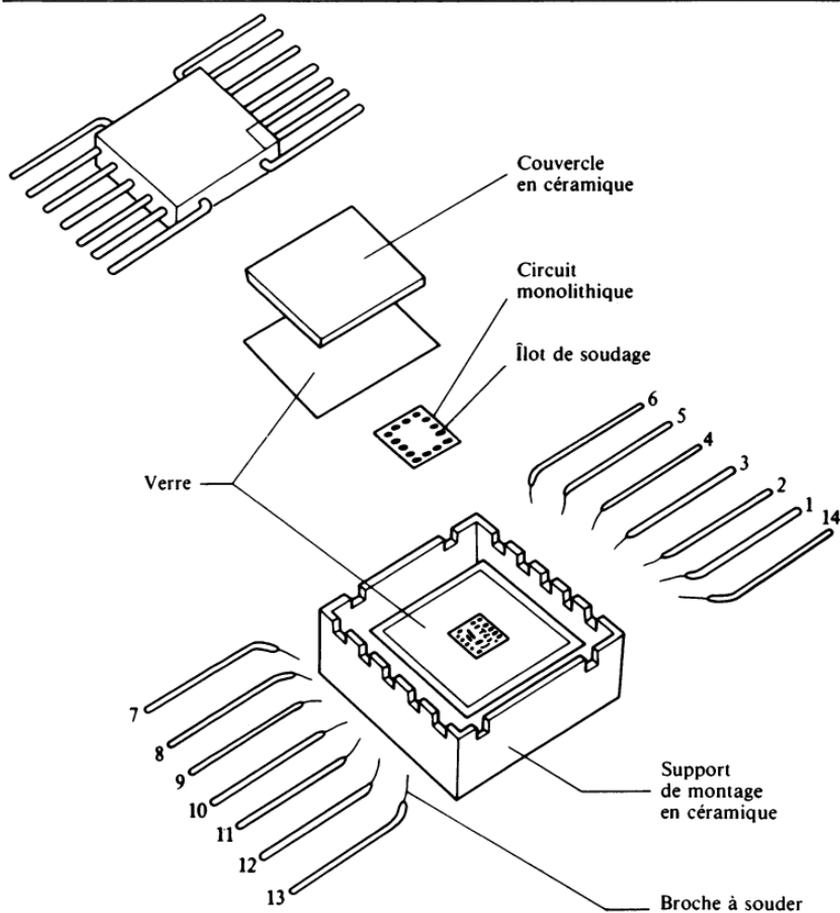


Fig. 14 Boîtier plat à 14 broches.

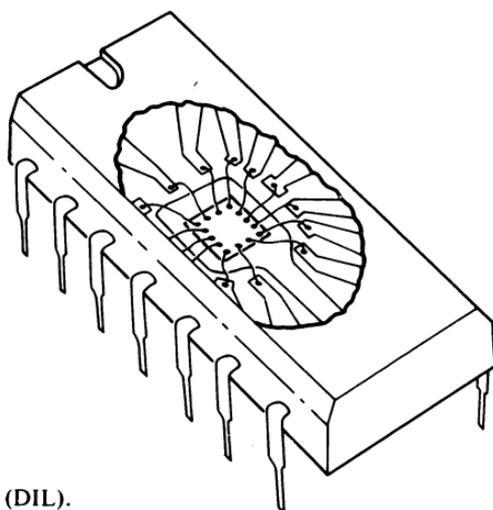


Fig. 15 Boîtier à double rangée (DIP).

pas international usuel, mais dispensait de l'obligation de réaliser des soudures supplémentaires. De plus, ce boîtier permettait un plus grand nombre de connexions avec l'extérieur (fig. 14).

• **boîtier plastique à double rangée de broches (DIL)**. Ce boîtier DIL a été développé pour répondre à un souci d'économie et pour accroître le nombre de connexions vers l'extérieur, puisque le nombre de broches est de 14, 16, 20, 24 ou 40. La figure 15 montre un boîtier DIL à 14 broches.

Abréviations d'usage courant :

ALU	Unité arithmétique et logique
BIT	Élément binaire
CCD	Dispositif à couplage de charge
CMOS	MOS complémentaire
CPU	Unité centrale de traitement
BCD	Décimal codé binaire
DIL	A double rangée de broches
DRL	Logique à diodes et résistances
DTL	Logique à diodes et transistors
EAROM	Mémoire morte altérable électriquement
ECL	Logique non saturée à couplage par les émetteurs
EPROM	Mémoire morte programmable et effaçable
FIFO	Premier entré, premier sorti
I ² L	Logique à injection de courant
LCD	Afficheur à cristaux liquides
LED	Diode électroluminescente
LIFO	Dernier entré, premier sorti
LSI	intégration à grande échelle (entre 100 et 1 000 portes)
MOS	Transistor métal-oxyde-semi-conducteur
MSI	Intégration à moyenne échelle (entre 10 et 100 portes)
NMOS	MOS à canal N
PROM	Mémoire morte programmable
RAM	Mémoire à accès direct
ROM	Mémoire morte
RTL	Logique à résistances et transistors
SOS	Silicium sur saphir
SSI	Intégration à faible échelle (< 10 portes)
TEC (FET)	Transistor à effet de champ
TTL	Logique tout à transistors
TTL H	Logique TTL rapide
TTL L	Logique TTL à faible consommation
TTL LS	Logique TTL Schottky à faible consommation
TTL S	Logique TTL Schottky
UJT	Transistor unijonction
VLSI	Intégration à très grande échelle (> 1 000 portes)
VMOS	MOS à structure verticale

de porte en porte

portes logiques de base

Les applications de la logique qui seront traitées ici se référeront à des systèmes simples de commande, d'alarme et de surveillance. Il est bien entendu que les méthodes énoncées pourront être étendues à des applications plus complexes, comme celles que l'on rencontre dans les ordinateurs, par exemple.

Les systèmes logiques se divisent en deux groupes :

Les systèmes combinatoires, dans lesquels une combinaison de plusieurs signaux d'entrée fournit instantanément un ou plusieurs signaux de sortie qui persistent tant que les signaux d'entrée sont appliqués;

Les systèmes séquentiels, dans lesquels une combinaison de plusieurs signaux d'entrée entraîne suivant l'ordre d'arrivée des signaux, un ou plusieurs signaux de sortie. Le temps intervenant dans les systèmes séquentiels, la mémorisation de ces signaux de sortie est possible, même après disparition des signaux d'entrée.

Ces systèmes peuvent être à nouveau classés en deux catégories : *les systèmes synchrones* dans lesquels les changements d'état de l'ensemble du système ne se font que sur réception d'un signal d'horloge qui synchronise le tout. Le signal d'horloge correspond à un train d'impulsions (*impulsions d'horloge*) fabriqué par un générateur d'horloge, complètement indépendant du système;

les systèmes asynchrones, dans lesquels les changements d'état ne sont pas synchronisés par un signal d'horloge; les sorties ne changent d'état que sur réception de signaux de commande.

L'algèbre de Boole

En 1854, Georges Boole écrivit un article intitulé « Investigation parmi les lois de la pensée ». Les mathématiques classiques, quoique extrêmement utiles dans la recherche intellectuelle, ne peuvent prendre en compte tous les aspects de la pensée.

L'exemple suivant montre l'inadaptation de l'algèbre classique à résoudre tous les problèmes :

Soient les deux suppositions suivantes :

- les chiens sont des animaux,
- les chats sont des animaux.

On en tire la conclusion absurde que tous les chiens sont des chats. Toutefois, si l'on considère le problème à l'aide de l'algèbre classique, on peut écrire :

$$A = B, C = B,$$

et donc $A = C$, qui est une déduction parfaitement correcte.

La difficulté inhérente à ce problème est une question de langage. Le signe mathématique de l'égalité est utilisé pour représenter le mot « sont ». Il est vrai qu'un chien est un animal, mais la réciproque n'est pas vraie (tous les animaux ne sont pas des chiens). En fait, le chien fait partie d'une sous-classe de la classe des animaux.

Boole développa une algèbre entièrement nouvelle qu'il appela « Algèbre des classes ». En dehors de l'intérêt que cela suscita à l'époque dans les milieux mathématiques, le traité de Boole resta dans la poussière des bibliothèques jusqu'en 1938. A cette époque, la technique téléphonique atteignait un haut degré de complexité, et les méthodes de Boole furent soudainement redécouvertes grâce à un article publié par C.E. Shannon et intitulé « Analyse symbolique des circuits de commutation à relais ».

Shannon découvrit que l'algèbre des classes de Boole était un outil puissant, qui permettait d'analyser et de représenter les circuits complexes, basés sur un fonctionnement à deux états.

Les règles de base de l'algèbre de Boole

- Une quantité ne peut avoir que deux valeurs : 0 ou 1. Il ne peut exister aucune autre valeur.
- Le sens de certains signes mathématiques est entièrement différent. Ainsi :

$A.B$ veut dire A et B , et non A que multiplie B .

$A + B$ veut dire A ou B et non A plus B .

\bar{A} veut dire « non A », c'est-à-dire le complément de a .

• Le signe de l'égalité prend une signification toute nouvelle et peut se définir comme suit :

= veut dire « *une sortie existe* » ou « *un interrupteur est fermé* ».

Les lois de la logique

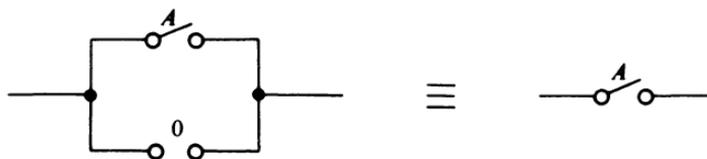
L'exactitude de certaines suppositions est évidente, alors que pour d'autres elle ne l'est pas du tout. Il est possible de vérifier la vraisemblance d'une supposition à partir du moment où elle est exacte, en utilisant la notation binaire. Ainsi, si la supposition est vraie, la fonction existe et vaut « 1 », et si la supposition est fausse, la fonction n'existe pas et vaut « 0 ».

- $A + 0 = A$
- $A.0 = 0$
- $A + 1 = 1$
- $A.1 = A$
- $A + A = A$
- $A.A = A$
- $A + \bar{A} = 1$
- $A.\bar{A} = 0$
- $\overline{A + B + C} = \bar{A}.\bar{B}.\bar{C}$
- $\overline{A.B.C} = \bar{A} + \bar{B} + \bar{C}$
- $(A + B).C = A.C + B.C$
- $A.B + C = (A + C).(B + C)$

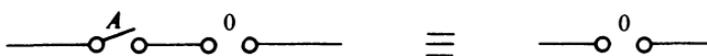
Les huit premières lois peuvent aisément se comprendre en examinant les arrangements des contacts de relais de la figure 1, dans lesquels un contact de travail (contact ouvert) est désigné par A et un contact de repos (contact fermé) est désigné par \bar{A} .

La représentation symbolique des éléments de logique

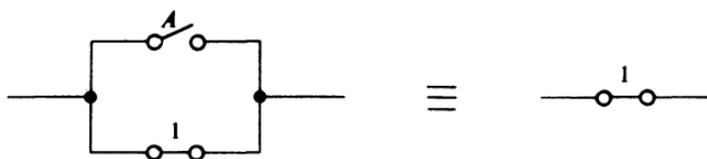
Les symboles graphiques utilisés dans les logigrammes, représentent des fonctions logiques. Ils sont indépendants de la technologie de réalisation de ces fonctions, qu'elle soit électronique, électromécanique, hydraulique, pneumatique, etc. Toutefois, dans le texte qui suit, ne sera considérée que la technologie électronique. Si le système électronique définit les deux états de la logique comme deux tensions (ou deux courants), et si la tension la plus positive correspond à l'état « 1 », le système fonctionne en *logique positive*. Dans le cas contraire, le système fonctionne en *logique négative*, si la tension la plus négative correspond à l'état « 1 ». Dans le cas où le système utilise indifféremment les deux logiques (positive et négative), on dit que le système fonctionne en logique mixte.



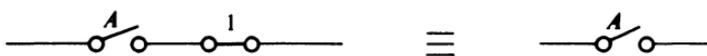
$$(a) A + 0 = A$$



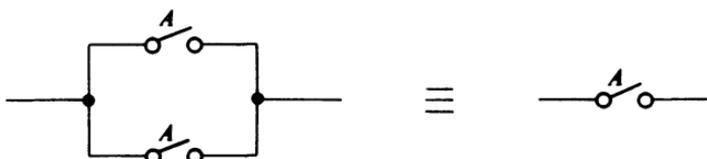
$$(b) A \cdot 0 = 0$$



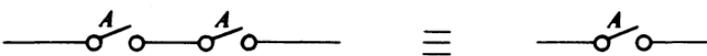
$$(c) A + 1 = 1$$



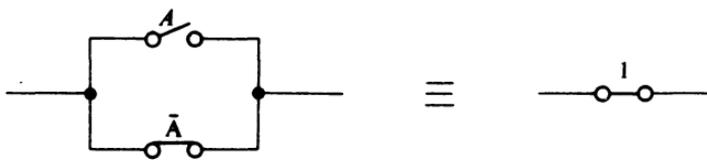
$$(d) A \cdot 1 = A$$



$$(e) A + A = A$$



$$(f) A \cdot A = A$$



$$(g) A + \bar{A} = 1$$



$$(h) A \cdot \bar{A} = 0$$

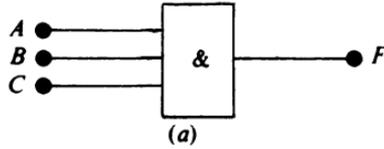
Fig. 1 Les lois de la logique.

Porte de base et tables de vérité

Porte ET (AND).

Le symbole logique est représenté à la figure 2 *a*. La sortie F vaut « 1 », lorsque les signaux d'entrée A et B et C valent tous « 1 ». La table de vérité donne les différentes valeurs de la fonction pour toutes les combinaisons des entrées (fig. 2). L'équation booléenne de la porte ET est alors :

$$F = A.B.C.$$



A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Fig. 2 Porte ET :
(a) symbole logique,
(b) table de vérité.

(b)

Porte OU (OR).

Le symbole logique est représenté à la figure 3 *a*. La sortie F vaut « 1 », lorsque l'un au moins des signaux d'entrée A ou B ou C vaut « 1 » (table de vérité en figure 3 *b*).

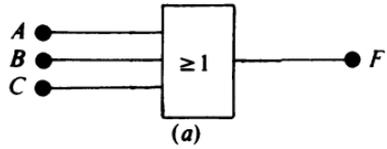
L'équation booléenne de la porte OR est alors :

$$F = A + B + C$$

Porte NON (NOT) (inverseur).

Le symbole logique est représenté à la figure 4 *a*. On notera le petit cercle symbolisant l'inversion, qui peut être placé avant ou après la porte, tel que l'indique la figure 4 *b*. Le signal de sortie vaut « 0 » quand le signal d'entrée vaut « 1 », et vice versa. L'équation booléenne de la porte NON est alors :

$$F = \bar{A}$$



A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

(b)

Fig. 3 Porte OU :
 (a) symbole logique,
 (b) table de vérité.

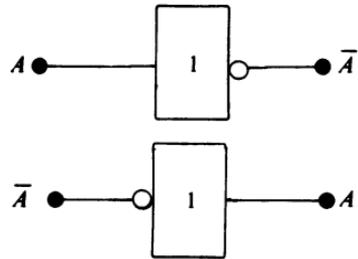


Fig. 4 Porte NON (inverseur).

Porte ET-NON (NAND).

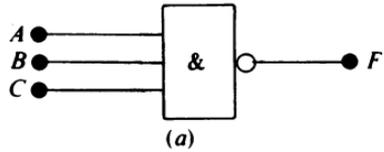
Le symbole logique est représenté à la figure 5 a. La sortie F vaut « 0 » lorsque les signaux d'entrée A et B et C valent tous « 1 » (fig. 5 b). L'équation booléenne de la porte ET-NON est alors :
 $F = \overline{A.B.C}$.

Porte OU-NON (NOR).

Le symbole logique est représenté à la figure 6 a. La sortie F vaut « 0 », lorsque l'un au moins des signaux d'entrée A ou B ou C vaut « 1 » (table de vérité en figure 6 b).

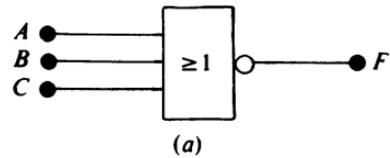
L'équation booléenne de la porte OU-NON est alors :

$$F = \overline{A + B + C}$$



A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Fig. 5 Porte ET-NON :
 (a) symbole logique,
 (b) table de vérité.



A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Fig. 6 Porte OU-NON :
 (a) symbole logique,
 (b) table de vérité

REMARQUE : Une porte ET en logique positive devient une porte OU en logique négative, et vice versa. De même, une porte ET-NON (NAND) en logique positive devient une porte OU-NON (NOR) en logique négative. Il suffit pour vérifier cela, de comparer les tables de vérité.

Systèmes de numération

Décimal. Ce système utilise 10 symboles pour représenter les quantités de 0 à 9. Le nombre de chiffres employés s'appelle la base. Ici, elle est égale à 10.

Les autres nombres sont construits en donnant différentes valeurs, ou *poids*, à la position relative des chiffres par rapport à la virgule décimale. Le poids de chaque position est une puissance de la base, ce qui donne dans le cas général :

$$R^2 R^1 R^0, R^{-1} R^{-2}$$

et pour le système décimal :

$$10^2 10^1 10^0, 10^{-1} 10^{-2}$$

EXEMPLE :

$$\begin{aligned} 426_{10} &= 4 \cdot 10^2 + 2 \cdot 10^1 + 6 \cdot 10^0 \\ &= 400 + 20 + 6 \\ &= 426 \end{aligned}$$

Octal. Ce système est très employé en informatique, car il est facile à convertir en binaire et vice versa. Son nombre plus réduit de chiffres le rend plus simple à utiliser que le binaire.

La base est égale à 8 et les poids relatifs sont des puissances de 8 :

$$8^2 8^1 8^0, 8^{-1} 8^{-2}$$

EXEMPLE :

$$\begin{aligned} 426_8 &= 4 \cdot 8^2 + 2 \cdot 8^1 + 6 \cdot 8^0 \\ &= 256 + 16 + 6 \\ &= 278_{10} \end{aligned}$$

Binaire. Ce système est le plus répandu en logique et en informatique. La base est égale à 2 et les poids relatifs sont des puissances de 2 :

$$2^4 2^3 2^2 2^1 2^0, 2^{-1} 2^{-2} 2^{-3}$$

Il existe diverses règles pour effectuer les additions, les soustractions, les multiplications et les divisions en binaire, ainsi que pour passer d'un système de numération à un autre. Certaines de ces règles sont abordées dans la suite (chapitre 8).

La comparaison entre ces trois systèmes de numération est faite dans le tableau de la figure 7.

Hexadécimal. Ce système est largement répandu dans l'affichage alphanumérique et en micro-informatique. Il utilise un nombre binaire à 4 chiffres pouvant représenter 16 symboles. Les dix premiers sont constitués par les chiffres décimaux de 0 à 9, et les 6 derniers sont constitués par les lettres de l'alphabet de A à F (fig. 8).

Décimal		Octal		Binaire				
10 ¹	10 ⁰	8 ¹	8 ⁰	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
	0		0					0
	1		1					1
	2		2				1	0
	3		3				1	1
	4		4			1	0	0
	5		5			1	0	1
	6		6			1	1	0
	7		7			1	1	1
	8	1	0		1	0	0	0
	9	1	1		1	0	0	1
1	0	1	2		1	0	1	0
1	1	1	3		1	0	1	1
1	2	1	4		1	1	0	0
1	3	1	5		1	1	0	1
1	4	1	6		1	1	1	0
1	5	1	7		1	1	1	1
1	6	2	0	1	0	0	0	0
1	7	2	1	1	0	0	0	1
1	8	2	2	1	0	0	1	0
1	9	2	3	1	0	0	1	1
2	0	2	4	1	0	1	0	0
2	1	2	5	1	0	1	0	1
2	2	2	6	1	0	1	1	0
2	3	2	7	1	0	1	1	1
2	4	3	0	1	1	0	0	0
2	5	3	1	1	1	0	0	1

Fig. 7 Comparaison entre les systèmes de numération.

Les nombres hexadécimaux dépassant la base peuvent être construits en prenant des groupes de 4 chiffres binaires pour chaque chiffre hexadécimal.

EXEMPLE :

0111 1001 représente 79

1011 0110 1101 1111 représente B6DF

Par conséquent, un tel système de numérotation est facile à interpréter.

Décimal	Binaire				Hexadécimal
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	1	0	1	0	A
11	1	0	1	1	B
12	1	1	0	0	C
13	1	1	0	1	D
14	1	1	1	0	E
15	1	1	1	1	F

Fig. 8 Système de numération hexadécimal.

des problèmes de famille

circuits intégrés numériques

Les éléments électroniques de logique ont évolué en passant par différents stades technologiques, dont le premier comportait des portes ET et OU à diodes. Le développement technologique permit ensuite de concevoir des portes à base de circuits actifs, et différentes familles logiques virent le jour. Les premiers éléments intégrés de logique ne furent que de simples transpositions des circuits à composants discrets. En fait, les tous premiers circuits étaient composés d'un assemblage de plusieurs puces de silicium raccordées entre elles par des fils. Au fur et à mesure du développement des techniques d'intégration, l'approche conceptuelle changea et les circuits commencèrent à être étudiés en fonction de la technologie de fabrication, et n'étaient plus les répliques intégrées d'un prototype à composants discrets. Dès que l'on comprit que la complexité d'un circuit n'était plus un facteur de limitation, la voie était ouverte à la production de circuits complexes à hautes performances.

Choix d'une famille logique

Le choix d'une famille logique pour une application donnée est généralement déterminé par les considérations suivantes :

- la vitesse de fonctionnement,
- l'immunité au bruit,

- l'entrée et la sortance,
- la consommation,
- la température de fonctionnement,
- le type de boîtier,
- le coût,
- la disponibilité.

Ces différents facteurs ne sont pas donnés nécessairement dans l'ordre; seules les applications envisagées donnent la priorité à tel ou tel facteur.

Vitesse de fonctionnement

La vitesse de fonctionnement d'une porte logique est définie par son temps de propagation, c'est-à-dire le temps que met le signal logique pour franchir cette porte.

En considérant une diode sur laquelle on inverse la tension aux bornes, passant d'une polarisation en direct à une polarisation en inverse, on peut illustrer l'un des facteurs contribuant à créer ce temps de propagation. Dans ce cas, le courant direct ne tombe pas instantanément à zéro, car les porteurs de charge doivent se recombinier en tout premier lieu avec les atomes, avant de disparaître. L'allure de l'impulsion de courant inverse est donnée par la figure 1, sur laquelle on voit la durée de décroissance jusqu'au

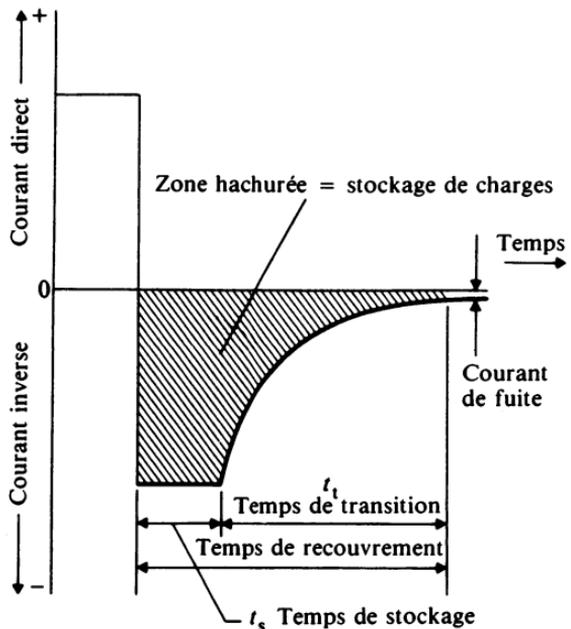


Fig. 1 Stockage de charges dans une diode.

courant de fuite. Pendant ce temps, la circulation du courant inverse oblige la jonction à stocker une charge électrique. Dès que les porteurs de charge ont été éliminés de la jonction (temps de stockage), le courant inverse décroît jusqu'à la valeur du courant de fuite (temps de décroissance). La durée totale (temps de recouvrement) représente le temps de propagation d'un signal commuté par une diode.

Les efforts accomplis pour améliorer les performances de commutation des dispositifs bipolaires ont conduit au développement des diodes Schottky, dans lesquelles une barrière existe entre un métal et un semi-conducteur de type n . La circulation de courant ne dépend plus des porteurs minoritaires, ce qui élimine pratiquement tout effet de stockage de charges. Les jonctions Schottky s'utilisent dans les diodes et les transistors dont les temps de propagation doivent être de l'ordre de la fraction de nanoseconde. Le symbole le plus répandu pour désigner une diode ou un transistor Schottky est illustré à la figure 2.

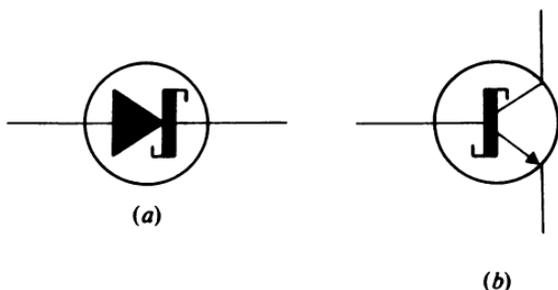


Fig. 2 Symboles des composants Schottky :

(a) diode Schottky,

(b) transistor Schottky.

Pour certains équipements numériques, comme les machines-outils à commande numérique, fonctionnant à faible vitesse, un temps de propagation de 1 ms est parfaitement acceptable. Pour d'autres équipements, par contre, comme les ordinateurs modernes, des temps de propagation de l'ordre de 1 ns sont nécessaires. La plupart des familles logiques intégrées actuellement fabriquées, ont des temps de propagation variant entre 2 ns et 100 ns. Les caractéristiques dissymétriques de commutation des circuits intégrés sont telles, qu'une transition d'un niveau logique bas à un niveau haut, présente un temps de transition qui est différent de celui d'une transition descendante. La valeur minimale du niveau haut et la valeur maximale du niveau bas sont généralement indiquées dans les spécifications de la famille logique considérée. Le temps de propagation « typique » est en principe la moyenne des deux temps de propagation (fig. 3, dans le cas d'un inverseur). Ces temps sont mesurés à mi-amplitude des signaux.

La capacité parasite présente à la sortie d'une porte, a une influence considérable sur le temps de propagation de celle-ci. C'est pourquoi on indique la valeur de cette capacité sur les figures représentant des transitions de signaux. Exemple : $C_p = 15 \text{ pF}$ ou 30 pF .

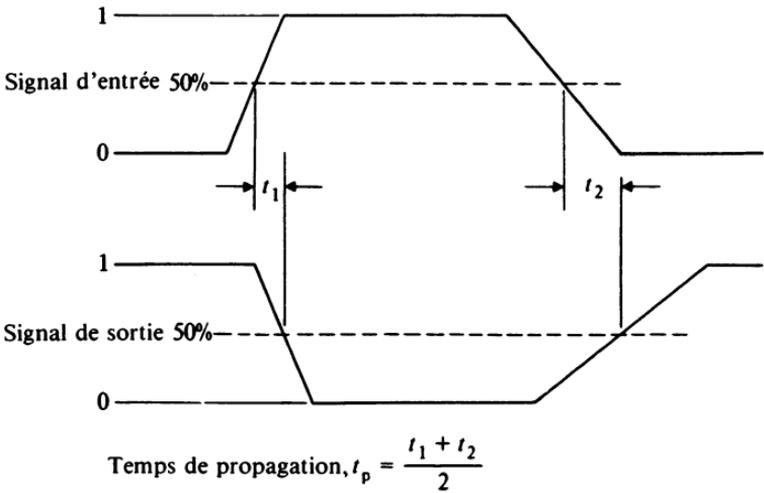


Fig. 3 Temps de propagation d'un inverseur (porte NON).

L'immunité au bruit

Toute tension parasite se superposant au signal utile par les interconnexions est considérée comme un bruit. Ce type de signal peut entraîner des commutations intempestives des portes. Le bruit provient généralement de l'intérieur d'un système car il résulte de la diaphonie existant entre les interconnexions. Les portes sont généralement conçues de façon à présenter une certaine immunité vis-à-vis du bruit, la marge du bruit étant définie par rapport aux niveaux haut et bas et au seuil d'entrée. Ainsi, la marge de bruit au niveau bas est égale à la différence entre la valeur maximale du niveau bas de sortie d'une porte et la valeur minimale du niveau de seuil d'entrée de cette porte. De la même façon, la marge de bruit au niveau haut est égale à la différence entre la valeur minimale du niveau haut de sortie d'une porte et la valeur maximale du niveau de seuil d'entrée de cette porte. Ainsi :

l'immunité au bruit est le degré avec lequel une porte peut supporter des variations du niveau d'entrée, sans que cela n'entraîne de modifications du niveau de sortie, ou en d'autres termes, la marge

de bruit est la différence entre la tension de sortie et la tension de seuil d'entrée :

En considérant la caractéristique de transfert d'une porte ET-NON (NAND) en technologie TTL (fig. 4), on constate que les zones hachurées correspondent aux valeurs interdites par les spécifications. Ainsi le niveau de sortie bas garanti est toujours inférieur à 0,4 V, le niveau de sortie haut garanti est toujours supérieur à 2,4 V ; le niveau de seuil garanti est compris entre 0,8 V et 2,0 V.

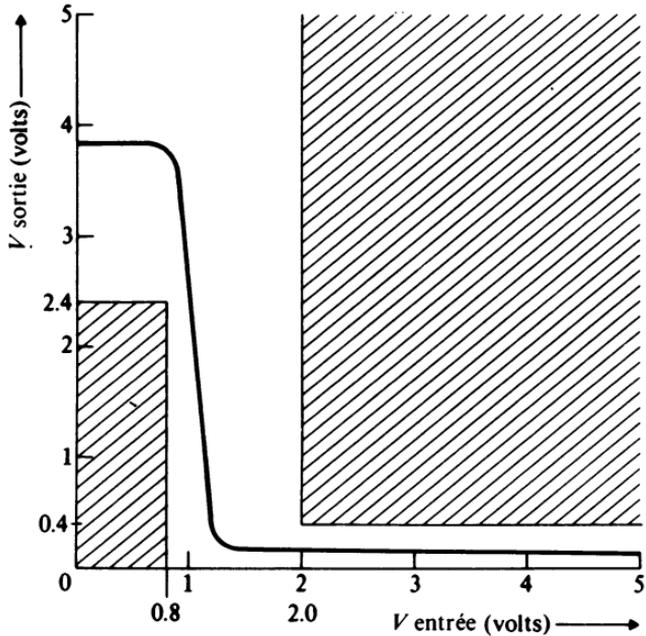


Fig. 4 Caractéristiques de transfert d'une porte ET-NON TTL.

La marge de bruit pour une porte ET-NON en TTL est alors, dans le plus mauvais des cas :

- au niveau bas = $0,8 - 0,4 = 0,4$ V
- au niveau haut = $2,4 - 2,0 = 0,4$ V
- marge de bruit garantie = $0,4$ V = 400 mV

L'entrance et la sortance

L'entrance d'une porte logique correspond au nombre maximal d'entrées indépendantes pour cette porte. Il est possible d'utiliser des expanseurs pour augmenter le nombre d'entrées d'une porte, mais apparaît rapidement une limitation : le temps de propagation de l'ensemble.

La *sortance* d'une porte logique correspond au nombre maximal d'entrées unitaires qu'elle peut alimenter simultanément, sans que les niveaux de sortie sortent des spécifications.

La consommation

Afin de déterminer la consommation globale d'un ensemble, il est nécessaire de connaître la consommation d'une porte logique. La consommation est différente, suivant que la porte délivre un niveau haut ou bas. La consommation typique d'une porte correspond à la moyenne de ces deux valeurs.

Les circuits rapides ont tendance à consommer davantage, car ils sont généralement conçus avec des valeurs de résistances plus faibles. De même, la charge de la capacité parasite draine d'autant plus de courant que la vitesse de commutation est rapide. La plupart des familles de circuits intégrés fonctionnent avec une tension d'alimentation de 5 V; leur consommation est comprise entre 1 mW et 100 mW par porte.

Température de fonctionnement

La température de fonctionnement correspond à la température ambiante dans laquelle le circuit intégré fonctionne correctement et respecte ses caractéristiques. Deux gammes de températures sont couramment utilisées : la gamme militaire ($- 55\text{ °C}$ à $+ 125\text{ °C}$) et la gamme commerciale (0 °C à $+ 70\text{ °C}$). Certaines familles ne sont disponibles que dans une plage restreinte de températures.

Les familles logiques

Les fabricants de circuits intégrés ont tendance à lancer sur le marché leurs propres familles logiques sans trop se préoccuper de standardisation. Toutefois, certains types de famille émergent, car les utilisateurs les préfèrent à d'autres. Tous les fabricants tendent à produire ces familles populaires, devenant ainsi d'autres sources d'approvisionnement pour l'utilisateur.

Les familles logiques de circuits intégrés sont usuellement classées en fonction de la configuration de la porte de base. Les principales familles sont les suivantes :

- logique à diodes et résistances (DRL),
- logique à résistances et transistors (RTL),
- logique à diodes et transistors (DTL),
- logique tout à transistors (TTL),
- logique non saturée à couplage par les émetteurs (ECL),
- logique à transistors MOS,
- logique à MOS complémentaires (CMOS).

La logique à diodes et résistances (DRL)

Porte ET à diodes.

On considère le circuit de la figure 5 a. Lorsque l'une au moins des entrées A ou B est raccordée au niveau logique « 0 » (0 V en logique positive), la ou les diodes d'entrée sont polarisées en direct et conduisent. Le courant traverse la résistance R , ce qui crée une chute de tension et détermine une tension de sortie voisine de 0 V en F . En réalité, la tension de sortie est égale à 0,7 V, car elle correspond à la chute de tension directe dans la diode au silicium (0,3 V pour une diode au germanium).

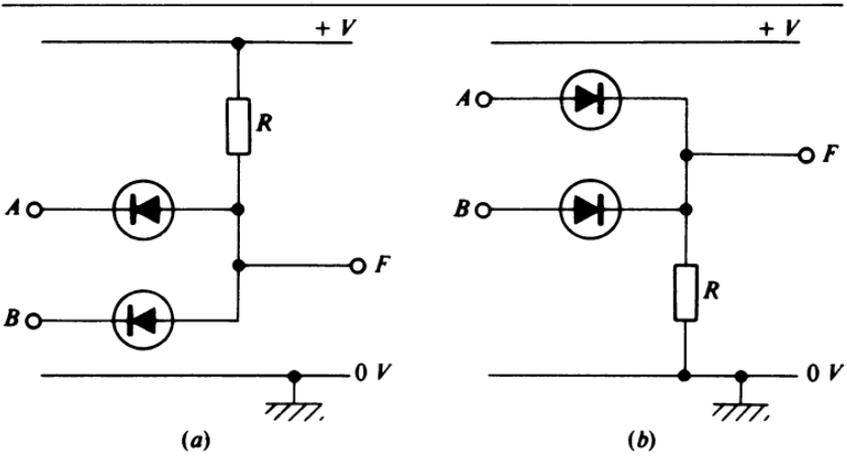


Fig. 5 Portes logiques à diodes et résistance (DRL) :

- (a) porte ET,
- (b) porte OU.

Lorsque les deux entrées A et B sont raccordées au niveau logique « 1 » (+ V en logique positive), les diodes d'entrée sont polarisées en inverse et ne conduisent pas (à l'exception d'un très faible courant de fuite). Dans ces conditions, la résistance R n'est qu'une liaison en continu entre la sortie F et la ligne d'alimentation + V; le niveau de sortie en F est égal à « 1 » (+ V).

Porte OU à diodes

On considère le circuit de la figure 5 b. Lorsque les deux entrées A et B sont reliées au niveau logique « 0 » (0 V), les diodes sont bloquées. Dans ces conditions, la résistance R n'est parcourue par aucun courant et le niveau de sortie en F est égal au « 0 » (0 V). Lorsque l'une au moins des entrées A ou B est raccordée au niveau logique « 1 » (+ V), la ou les diodes d'entrée sont polarisées en direct et conduisent. Le courant traverse la résistance R , créant une chute de tension, et détermine une tension de sortie au point F voisine de + V. En réalité, la tension de sortie est égale à $V - V_F$

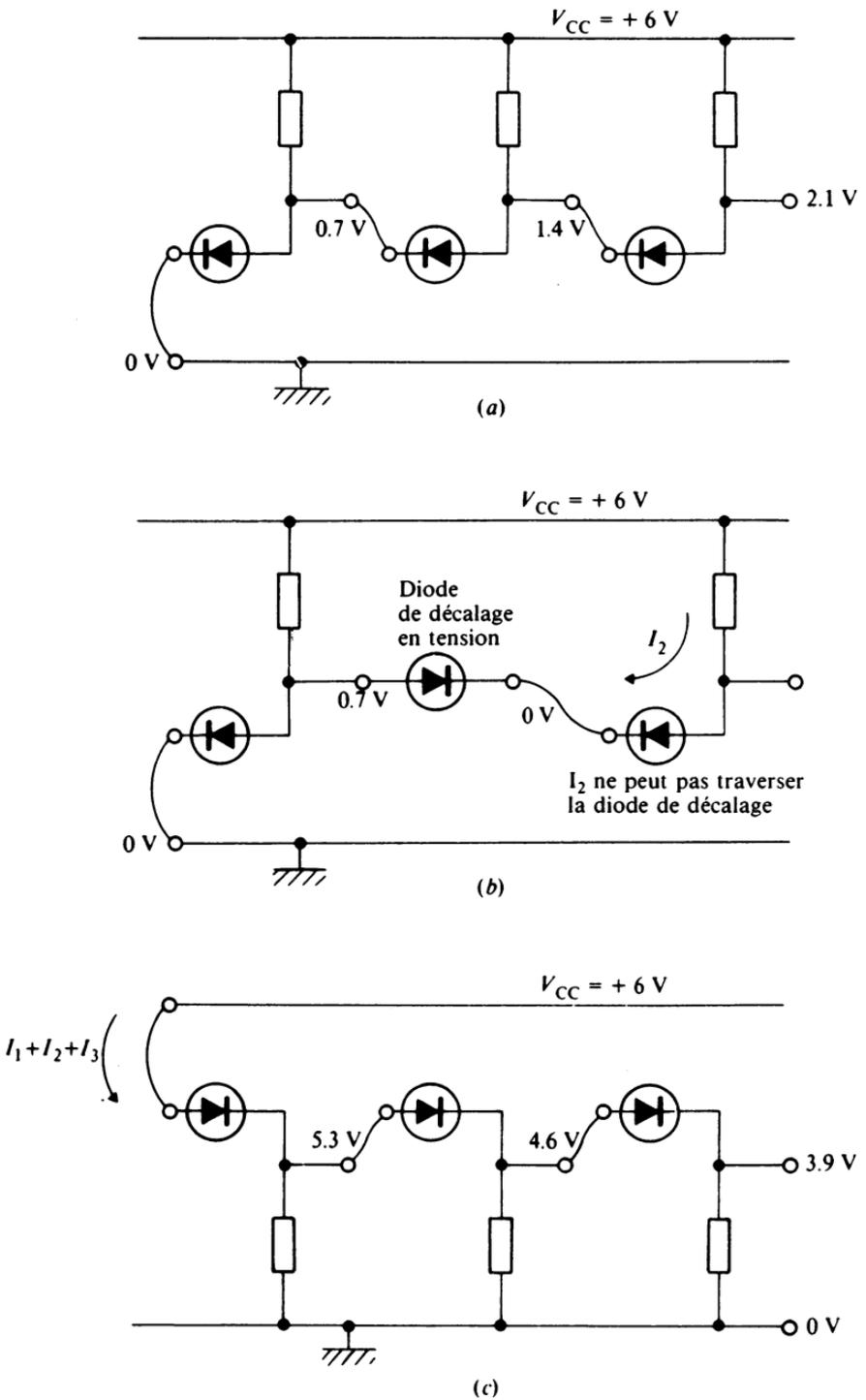


Fig. 6 Portes à diodes en cascade :

- (a) portes ET en cascade,
- (b) portes ET avec diode de décalage
- (c) portes OU en cascade.

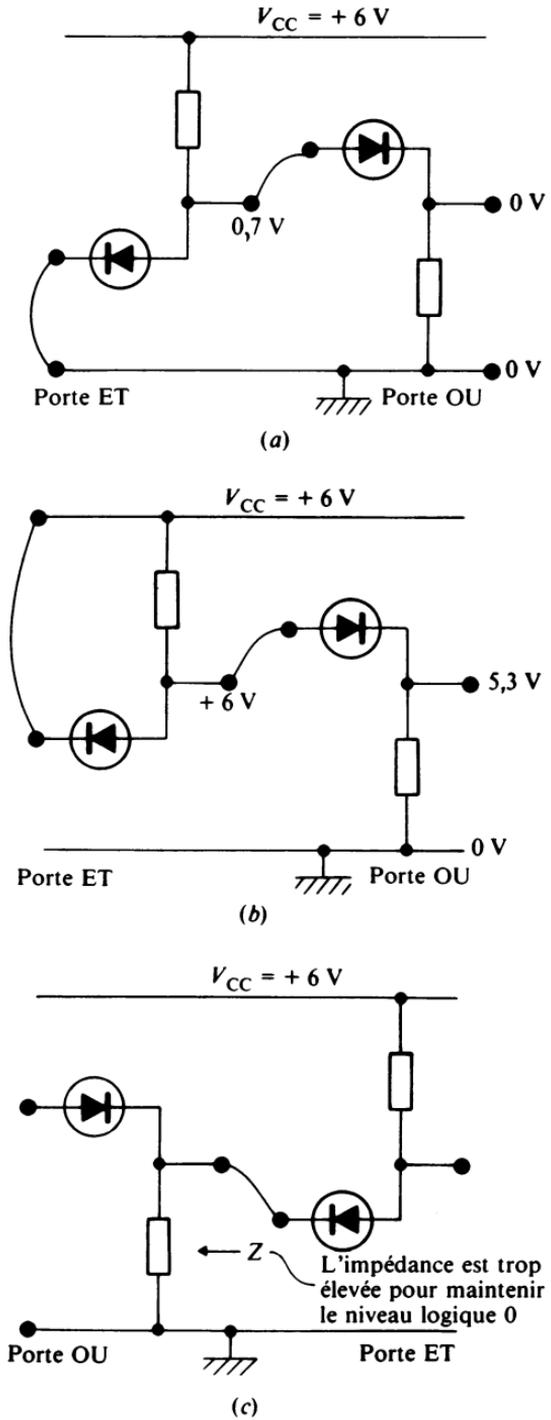


Fig. 7 Portes de commande à diodes :

- (a) porte ET commandant une porte OU au niveau bas,
- (b) porte ET commandant une porte OU au niveau haut,
- (c) porte OU ne pouvant commander une porte ET.

(V_F correspond à chute de tension directe dans la diode : 0,7 V si la diode est au silicium, 0,3 V pour une diode au germanium).

Portes à diodes en cascade

Lorsque des portes ET à diodes sont mises en cascade, il est nécessaire de s'assurer que le décalage en tension de l'étage final (dû aux chutes de tension directes dans les diodes mises en cascade) ne fausse pas la valeur de la tension de sortie (fig. 6 a). Il est possible d'utiliser une diode qui, polarisée en direct, supprime la tension de décalage, mais interdit de raccorder d'autres portes ET sur la sortie de cette porte ET. En effet, le courant I_2 de la porte ET rajoutée entraînerait (à l'état bas) une polarisation en inverse la diode de décalage (fig. 6 b).

Lorsque des portes OU à diodes sont mises en cascade, le niveau logique « 1 » (+ V) de chaque étage est réduit de la valeur de la chute de tension directe dans la diode. De plus, le signal d'entrée de la première porte doit fournir le courant de chaque porte, comme le montre la figure 6 c.

Il est possible d'utiliser des portes ET à diodes pour commander des portes OU à diodes (fig. 7 a et 7 b), mais l'inverse est impossible : une porte OU à diodes ne peut pas commander une porte ET à diodes, car l'impédance de la porte OU est trop élevée pour maintenir le niveau logique « 0 » (0 V) (fig. 7 c).

MONTAGE PRATIQUE

Logique à diodes et résistances

- Réaliser les deux montages des figures 8 a et 8 b.
- Appliquer toutes les combinaisons possibles de signaux logiques aux entrées A et B et dresser la table de vérité de ces deux portes. Vérifier également la qualité des niveaux logiques 0 et 1.

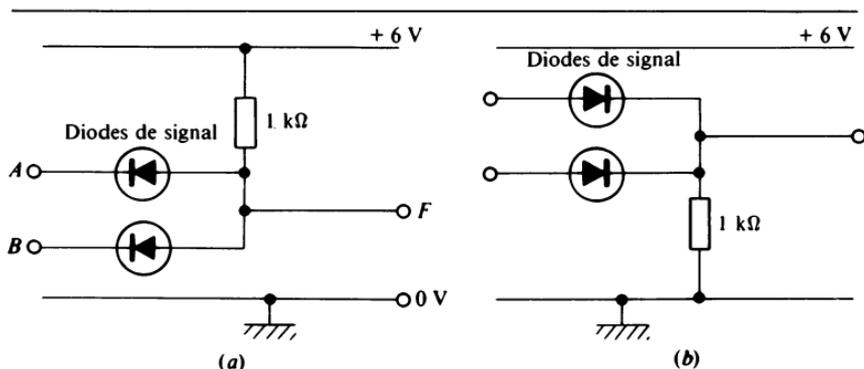


Fig. 8 Portes à diodes :

- (a) porte ET à diodes,
 (b) porte OU à diodes.

- Vérifier la sortance de chaque porte en chargeant la sortie par un ensemble de diodes et de résistances simulant des portes à diodes (on peut aller jusqu'à 6 portes).
- Réaliser les trois montages des figures 7 a, 7 b et 7 c, afin de vérifier les possibilités de commande de chaque type de porte.

La logique à résistances et transistors (RTL)

Cette logique fut la première à être intégrée. A cette époque, les fabricants choisirent d'intégrer une famille de circuits logiques qui avait fait ses preuves en composants discrets. Le choix se porta sur la logique RTL (logique à résistances et transistors) dont un exemple de porte OU-NON est donné à la figure 9.

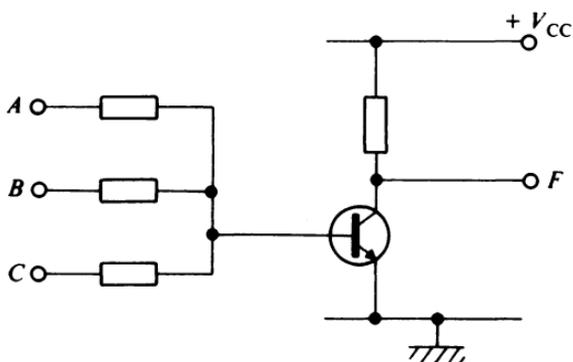


Fig. 9 Porte OU-NON RTL.

Il est préférable que les circuits à intégrer présentent le moins de composants passifs possibles, pour faciliter leur fabrication. Or, la famille logique RTL présente un grand nombre de résistances, vis-à-vis du nombre de transistors employés. De plus, la logique RTL présente une faible sortance, accompagnée d'une faible immunité au bruit (de l'ordre de 300 mV). Par contre, ce type de porte possède une vitesse de fonctionnement assez rapide pour une consommation raisonnable, puisque des temps de propagation de 12 à 40 ns sont possibles avec une consommation comprise entre 20 et 2 mW.

MONTAGE PRATIQUE

Porte RTL à composants discrets

- Réaliser le montage de la figure 10.
- Appliquer toutes les combinaisons possibles de signaux logiques aux entrées A et B, et dresser la table de vérité de cette porte. Vérifier également la qualité des niveaux logiques 0 et 1.

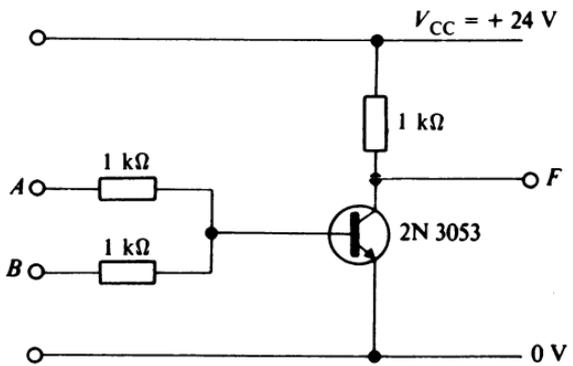


Fig. 10 Porte OU-NON RTL à composants discrets.

- Mesurer la valeur du courant de commande du transistor. Ce courant correspond à une unité d'entrée pour la logique RTL.
- Vérifier la sortance de cette porte en chargeant sa sortie par d'autres portes, jusqu'à ce que la tension du point F tombe à 18 V (les entrées A et B étant au niveau logique « 0 »). Déterminer alors la sortance de cette porte. Il suffit pour cela de diviser le courant qui traverse la charge par la valeur du courant unitaire trouvée précédemment.

Comment augmenter la sortance de cette porte?

REMARQUE : Avec des résistances de base de 10 kΩ, le courant de base est plus que suffisant pour saturer le transistor ($V_{CE\text{ sat}} \approx 0,2\text{ V}$ pour un transistor de commutation au silicium). Dans le circuit d'entrée de cette porte, on trouve une résistance de 10 kΩ en série avec la base d'un transistor monté en émetteur commun. En négligeant la chute de tension base-émetteur, on peut donc considérer que la résistance d'entrée de cette porte est de 10 kΩ. Ainsi, pour déterminer la sortance de cette porte, il suffit de brancher en parallèle des résistances de 10 kΩ entre le point F et la masse, le nombre de résistances utilisées pour faire chuter la tension de sortie jusqu'à 18 V, correspond au nombre de portes RTL que cette porte peut commander.

En rajoutant en série sur une des entrées un potentiomètre de 1 MΩ, le courant minimum de base permettant de saturer le transistor peut être mesuré; cela permet de déterminer la nouvelle valeur de sortance que l'on pourrait obtenir, en augmentant ainsi les résistances d'entrée des portes RTL.

Logique à diodes et transistors (DTL)

Cette famille logique fut la première série vraiment populaire de circuits intégrés; elle est encore parfois utilisée de nos jours. Une

porte élémentaire ET-NON en technologie DTL (logique à diodes et transistors) est représentée à la figure 11. Ce montage simple comprend une porte ET à diodes, deux diodes de décalage de tension et un transistor d'amplification et d'inversion. Ce montage peut être amélioré en ajoutant un transistor pour diminuer le courant dans les diodes d'entrée et donc augmenter la sortance de cette porte.

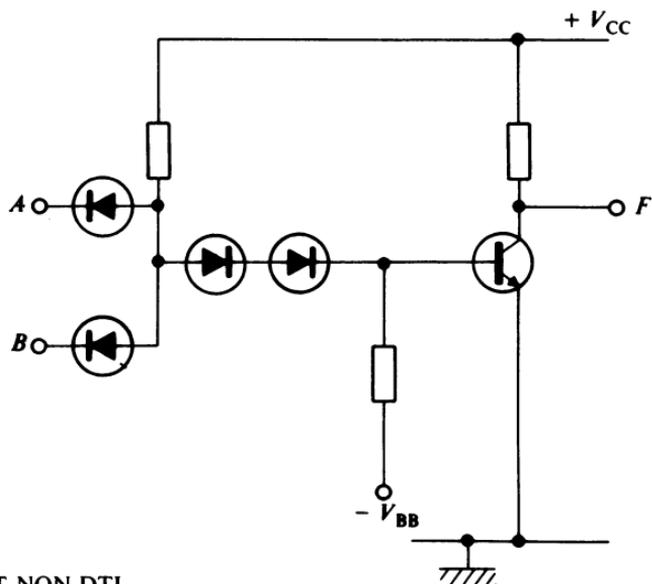


Fig. 11 Porte ET-NON DTL.

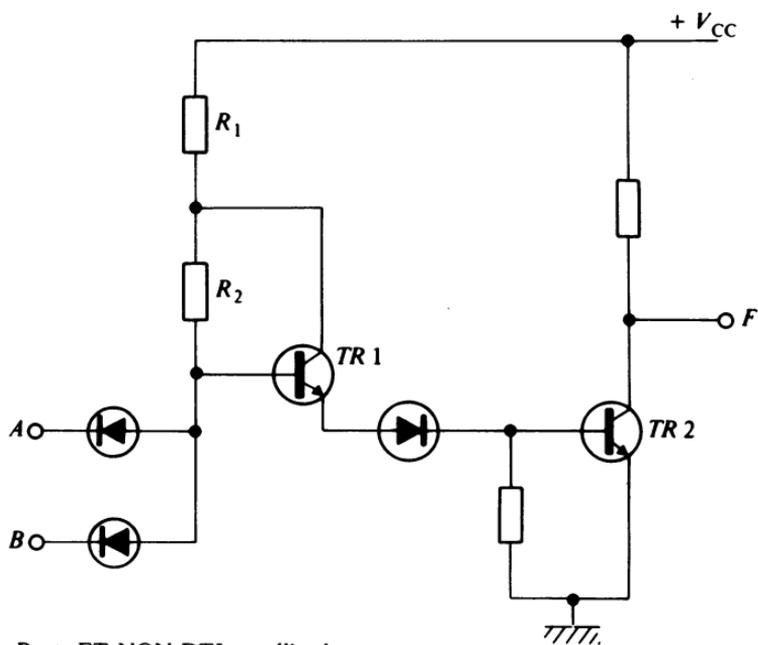


Fig. 12 Porte ET-NON DTL améliorée.

Lorsque les signaux d'entrée sont au niveau « 1 », les diodes d'entrée sont polarisées en inverse et tout le courant qui traverse les résistances R_1 et R_2 , sature $TR1$ qui, par la diode de décalage et la base sature $TR2$ et conduit le point F de sortie au niveau logique « 0 ».

Lorsque l'une au moins des entrées est au niveau « 0 », la diode correspondante dérive le courant provenant de l'alimentation au travers de R_1 et R_2 , provoquant le blocage des transistors $TR1$ et $TR2$ et conduisant la sortie F au niveau « 1 ». Cette porte réalise donc la fonction ET-NON (NAND).

Les circuits intégrés DTL ont des temps de propagation moyens de 25 ns avec une consommation par porte comprise entre 5 et 10 mW, et une sortance comprise entre 8 et 10. L'immunité au bruit est d'environ 1 V. Une variante de circuits intégrés DTL existe, ayant des temps de propagation de 50 ns à 60 ns, une consommation de 1 à 2 mW par porte et une immunité au bruit de 5 V.

MONTAGE PRATIQUE

Porte DTL à composants discrets.

- Réaliser le montage de la figure 13.
- Appliquer toutes les combinaisons possibles de signaux logiques aux entrées A et B , et dresser la table de vérité de cette porte. Vérifier également la qualité des niveaux logiques « 0 » et « 1 ».

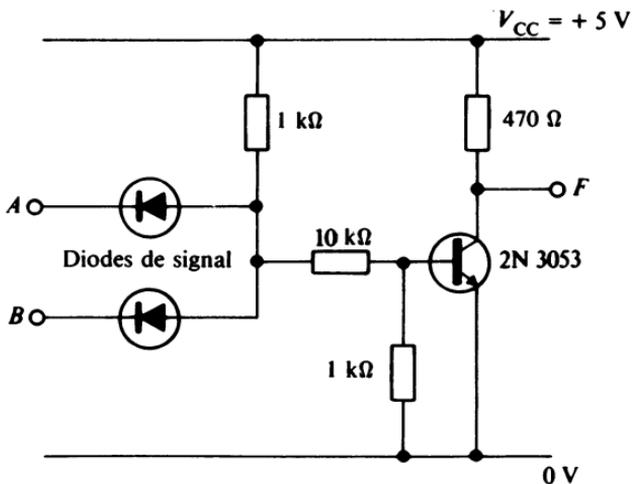


Fig. 13 Porte ET-NON DTL à composants discrets.

- Vérifier la sortance de cette porte en la chargeant par d'autres portes.

La logique tout à transistors (TTL)

On peut considérer la logique TTL (logique tout à transistors), comme une amélioration de la logique DTL. Les diodes d'entrée et la diode de décalage ont été remplacées par un transistor multi-émetteur, comme indiqué sur la figure 14. La fabrication de ce transistor multi-émetteur est facilitée par les techniques d'intégration.

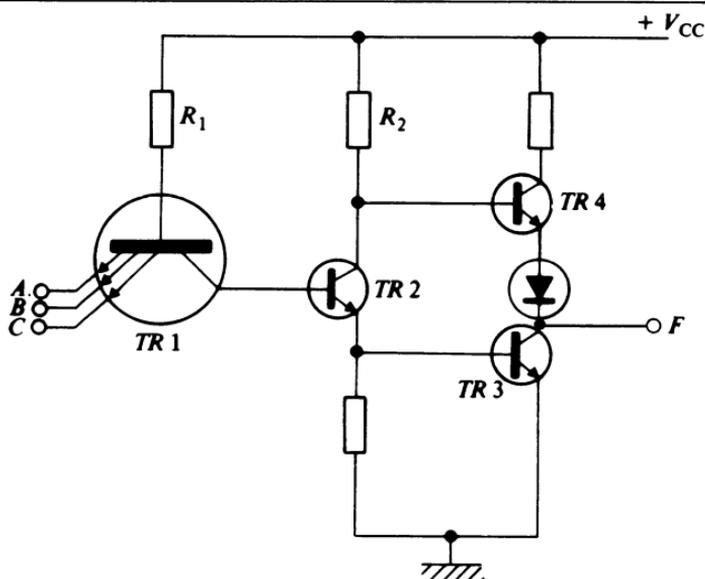


Fig. 14 Porte ET-NON TTL.

Lorsque toutes les entrées sont au niveau logique « 1 », la jonction base-émetteur de $TR1$ est polarisée en inverse; le courant qui provient de l'alimentation à travers R_1 et la jonction base-collecteur de $TR1$ rend fortement passant $TR2$, ce qui sature $TR3$ (et bloque $TR4$), de façon à fournir au point F un niveau logique « 0 ». L'étage de sortie est une forme d'amplificateur symétrique série appelé « totem pole » (à transistors NPN).

Lorsque l'une au moins des entrées est au niveau « 0 », la jonction base-émetteur correspondante de $TR1$ est parcourue par le courant provenant de l'alimentation par R_1 . La chute de tension entre la base de $TR1$ et la masse est alors insuffisante pour rendre passant la jonction base-collecteur de $TR1$ et la jonction base-émetteur de $TR2$; ceci bloque $TR3$ et rend passant $TR4$ qui se sature. La sortie F est au niveau « 1 ».

La logique TTL est devenue extrêmement populaire, et plusieurs versions se sont développées afin d'améliorer certains paramètres, comme le temps de propagation, la consommation ou l'immunité

au bruit, dont les valeurs moyennes sont respectivement les suivantes pour les différentes familles :

- TTL standard : 10 ns, 10 mW et 1 V,
- TTL H (rapide) : 6 ns, 10 mW et 1 V,
- TTL L (faible consommation) : 35 ns, 1 mW et 1 V,
- TTL S (Schottky) : 3 ns, 10 mW et 0,9 V,
- TTL LS (Schottky à faible consommation) : 10 ns, 2 mW et 0,8 V.

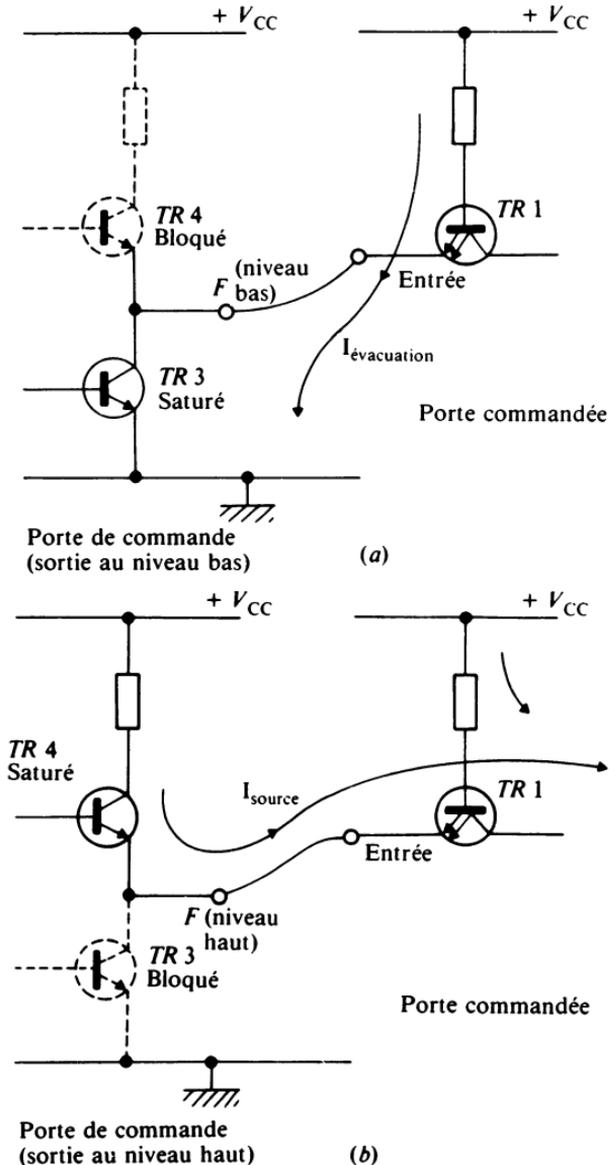


Fig. 15 Courants d'évacuation et de source d'une porte TTL :

- (a) courant d'évacuation,
 (b) courant de source.

Courants d'évacuation et de source

Lorsqu'une porte TTL commande d'autres portes TTL, les courants limites de commande sont spécifiés par le fabricant. Ainsi, lorsque la sortie d'une porte de commande est au niveau bas (« 0 » logique), la circulation du courant est indiquée à la figure 15 *a*; on dit que la porte de commande *évacue* ce courant à la masse. Les spécifications indiquent que le courant d'évacuation d'entrée d'une porte est de 1,6 mA au maximum. Comme chaque sortie de porte peut évacuer un courant de 16 mA, la sortance à niveau bas d'une telle porte est 10. De même, lorsque la sortie d'une porte de commande est au niveau haut (« 1 » logique), la circulation du courant est celle indiquée à la figure 15 *b*; on dit que la porte de commande fournit un courant de *source*. Les spécifications indiquent que le courant de source d'entrée d'une porte est de 40 μA au maximum. Comme chaque sortie peut fournir un courant de 400 μA à l'état haut, la sortance à niveau haut est aussi 10.

REMARQUE : Par convention, dans les spécifications tout courant *entrant* dans le circuit est considéré comme positif et tout courant *sortant* du circuit, comme négatif.

EXEMPLE : Pour une porte TTL :

- entrée à l'état « 0 » : $I_c = - 1,6 \text{ mA}$
- entrée à l'état « 1 » : $I_c = 40 \mu\text{A}$
- sortie à l'état « 0 » : $I_s = 16 \text{ mA}$
- sortie à l'état « 1 » : $I_s = - 400 \mu\text{A}$

MONTAGE PRATIQUE

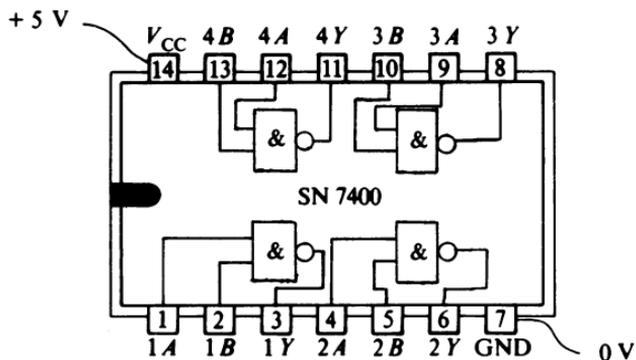
Porte ET-NON (NAND) en TTL

On utilise le circuit intégré SN 7400 contenant 4 portes ET-NON à deux entrées (fig. 16 *a*).

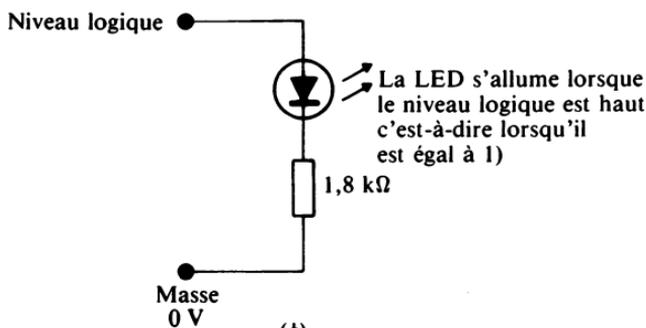
- Brancher respectivement l'alimentation au + 5 V et la masse aux broches 14 et 7.
- Appliquer toutes les combinaisons possibles de signaux logiques aux entrées des portes et vérifier l'état correspondant des sorties en utilisant l'indicateur à LED de la figure 16 *b*. La résistance de 1,8 k Ω réduit l'éclairage de la LED mais elle évite de trop charger une sortie de porte. Il faut se rappeler que :

le « 1 » logique correspond au + 5 V,

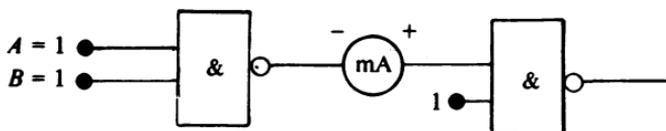
le « 0 » logique correspond au 0 V.



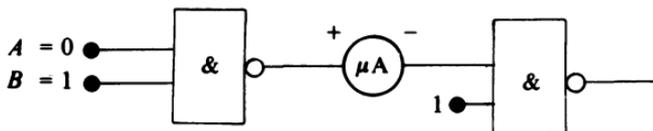
(a)



(b)



(c)



(d)

Fig. 16 Portes ET-NON TTL (SN 7400) :

- (a) quadruple porte ET-NON à 2 entrées (vue de dessus),
- (b) indicateur d'états logiques
- (c) montage de mesure du courant d'évacuation,
- (d) montage de mesure du courant de source.

Lorsque l'on utilise des portes ET-NON, il faut réunir les entrées inutilisées aux entrées utilisées, ou les relier au + 5 V à travers une résistance de 1 kΩ. Toutefois, ici, et de manière temporaire, les

entrées inutilisées peuvent être directement reliées au + 5 V sans risquer de dégâts.

- *Vérifier également la qualité des niveaux logiques « 0 » et « 1 », en se servant d'un multimètre numérique.*

REMARQUE : Il peut être nécessaire d'inverser les fils du multimètre numérique, pour obtenir une lecture convenable. Ce défaut peut se produire avec certains appareils utilisant un circuit CMOS, lorsqu'on effectue des mesures de tension à haute impédance. Certaines tensions parasites apparaissent dans le circuit de masse de l'appareil de mesure, perturbant ainsi la lecture. Il est évident que l'inversion des fils inverse également la polarité affichée.

- *Mesurer le courant d'évacuation au niveau bas et le courant de source au niveau haut, avec le montage de la figure 16 c. Vérifier que ces courants sont respectivement inférieurs à 1,6 mA et 40 μ A.*

La désignation des circuits TTL

Chaque élément du code de désignation renseigne sur le composant. Ainsi, pour le circuit SN 74H107 N, par exemple, le code de désignation fournit les indications suivantes :

- SN : réseau à semi-conducteurs.
- 74 : les circuits TTL sont fabriqués pour supporter deux gammes de températures.

La série militaire 54 supporte les températures comprises entre $- 55^{\circ}\text{C}$ et $+ 125^{\circ}\text{C}$.

La série industrielle 74 supporte les températures comprises entre 0°C et $+ 70^{\circ}\text{C}$.

Les tolérances sur la tension d'alimentation sont différentes suivant la série.

La série 54 admet des tensions d'alimentation comprises entre 4,5 V et 5,5 V ($\pm 10\%$).

La série 74 admet des tensions d'alimentation comprises entre 4,75 V et 5,25 V ($\pm 5\%$).

- H : circuit rapide. Les différentes variantes sont indiquées par les lettres suivantes :

L (faible consommation), S (Schottky), LS (Schottky à faible consommation).

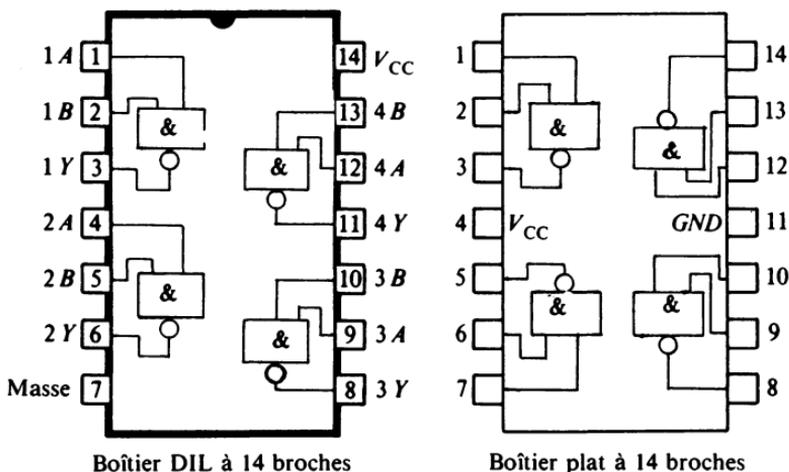
L'absence de lettre correspond à la série TTL standard.

- 107 : fonction du circuit (indiquée par deux ou trois chiffres). Le circuit 107 est une double bascule JK.

- N : type de boîtier (indiqué par une lettre) : W (boîtier plat), J (boîtier DIL céramique), N (boîtier DIL plastique), (le plus utilisé).

Feuille de caractéristiques d'un circuit TTL

Une feuille de caractéristiques typiques d'une quadruple porte ET-NON à 2 entrées est donnée à la figure 17. Ces feuilles couvrent les deux séries 54 et 74, aux différences près énumérées au paragraphe précédent.



Tension max. d'alimentation : $V_{CC} = 7 \text{ V}$

Tension d'entrée max. : $V_{IN} = 5,5 \text{ V}$

Tension nominale d'alimentation, série 54 : $V_{CC} = 5 \text{ V} \pm 0,5 \text{ V}$

Tension nominale d'alimentation, série 74 : $V_{CC} = 5 \text{ V} \pm 0,25 \text{ V}$

Température ambiante, série 54 : $T_A = -55 \text{ }^\circ\text{C}$ à $+125 \text{ }^\circ\text{C}$

Température ambiante, série 74 : $T_A = 0 \text{ }^\circ\text{C}$ à $+70 \text{ }^\circ\text{C}$

Caractéristiques électriques :

Tension d'entrée, niveau bas : $V_{E \text{ bas}} = 0,8 \text{ V max.}$

Tension d'entrée, niveau haut : $V_{E \text{ haut}} = 2,0 \text{ V min.}$

Tension de sortie, niveau bas : $V_{S \text{ bas}} = 0,4 \text{ V max. (typ. } 0,22 \text{ V)}$

Tension de sortie, niveau haut : $V_{S \text{ haut}} = 2,4 \text{ V min. (typ. } 3,3 \text{ V)}$

Courant d'entrée, niveau bas : $I_{E \text{ bas}} = -1,6 \text{ mA}$

Courant d'entrée, niveau haut : $I_{E \text{ haut}} = 40 \text{ } \mu\text{A}$

Caractéristiques de commutation :

Charge standard : $R = 400 \text{ } \Omega$, $C = 15 \text{ pF}$

Temps de propagation (transition montante) : $T_{PBH} = 22 \text{ ns max. (typ. } 11 \text{ ns)}$

Temps de propagation (transition descendante) : $T_{PHB} = 15 \text{ ns max. (typ. } 7 \text{ ns)}$

Fig. 17 Feuille de spécification d'un circuit TTL type SN 7400.

Le brochage de boîtiers à double rangée de broches (DIL) et des boîtiers plats y figurent vue de dessus. La numérotation et l'affectation des broches est différente pour les deux types de boîtiers. Les feuilles de caractéristiques comprennent les valeurs normales de fonctionnement, les gammes de températures et la sortance, les caractéristiques limites, abordées précédemment, les caractéristiques de commutation associées aux temps de propagation, suivant qu'il s'agit d'une transition montante ou descendante. Ces carac-

téristiques sont généralement établies à partir d'une charge standard constituée d'une résistance de 400Ω et d'un condensateur de 15 pF .

Logique non saturée à couplage par les émetteurs (ECL)

Le principe de fonctionnement de cette famille logique diffère du précédent, car les transistors ne fonctionnent pas en mode saturé. Cette famille fut développée pour obtenir les vitesses de fonctionnement plus rapides requises par les ordinateurs modernes. La figure 18 montre la structure d'une porte ECL type qui réalise simultanément les fonctions OU et OU-NON (NOR).

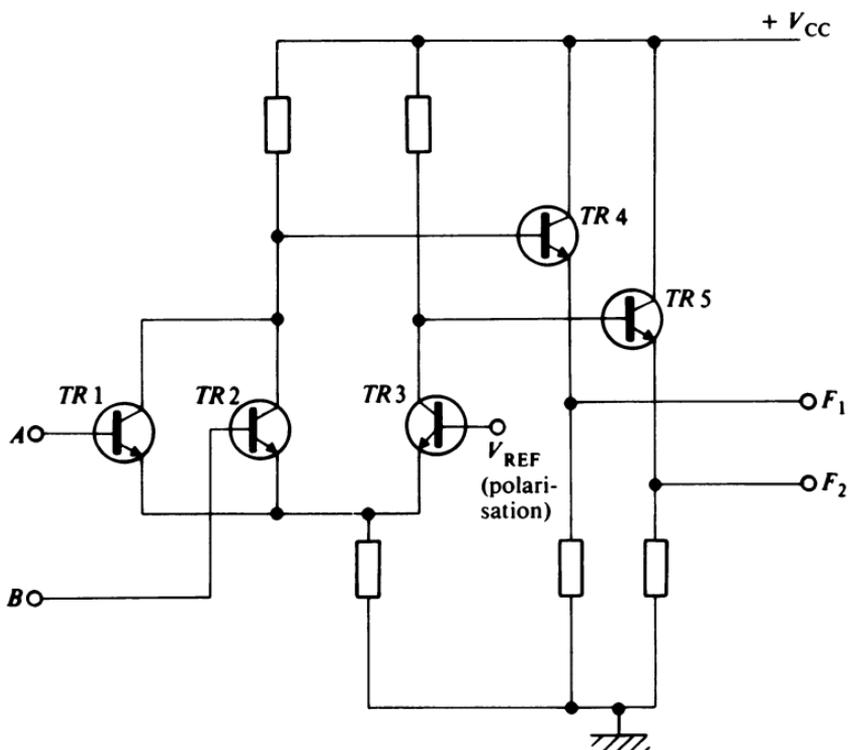


Fig. 18 Porte OU/OU-NON ECL.

Lorsque les deux entrées A et B sont au niveau bas, $TR1$ et $TR2$ sont bloqués et leur tension de collecteur est élevée. Le transistor $TR3$ devient alors passant grâce à la tension de polarisation V_{REF} appliquée à sa base, et sa tension de collecteur est faible. Les deux transistors $TR4$ et $TR5$, montés en collecteur commun, reproduisent fidèlement, à la tension base-émetteur près ($0,7 \text{ V}$), les tensions de collecteur de $TR1$, $TR2$ et de $TR3$. La sortie F_1 est donc au niveau haut et la sortie F_2 est au niveau bas. (La faible tension

collecteur de $TR3$ ne peut faire conduire $TR5$.) Lorsque l'une quelconque des entrées est au niveau haut, $TR1$ ou $TR2$ deviennent passants et leur tension de collecteur est faible. Le transistor $TR3$ est alors bloqué et sa tension de collecteur est élevée. La sortie F_1 est donc au niveau bas et la sortie F_2 au niveau haut. Par conséquent, la sortie F_2 réalise la fonction OU et la sortie F_1 réalise la fonction OU-NON.

Des temps de propagation typiques de 2 ns sont atteints avec une consommation de 25 mW par porte. La sortance élevée est égale à 30, mais l'immunité au bruit est très faible (200 mV). Il existe même des circuits ECL atteignant un temps de propagation de 1 ns.

Logique à transistors MOS

Les avantages principaux de la technologie MOS résident dans la faible surface d'intégration nécessaire et la très faible consommation des circuits. Ces avantages conduisent d'emblée à l'intégration à grande échelle (LSI) utilisée dans les mémoires d'ordinateur. Une porte élémentaire NON à transistors MOS à canal p est représentée en figure 19. Le transistor MOS 2 se comporte comme

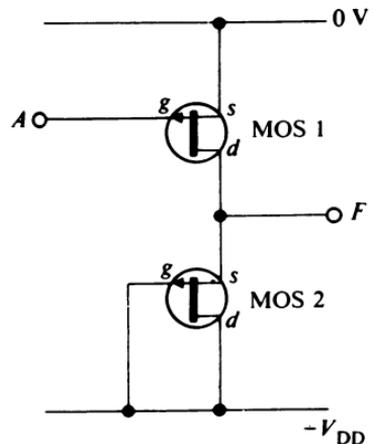


Fig. 19 Porte NON MOS.

une résistance de charge de drain pour le transistor MOS 1. Lorsque le signal d'entrée est au niveau bas (tension négative pour un MOS à canal p), le MOS 1 est saturé, et la sortie F est au niveau haut (0 V). Lorsque le signal d'entrée est au niveau haut, le MOS 1 est bloqué et la sortie F est au niveau bas ($-V_{DD}$) procurant ainsi l'inversion.

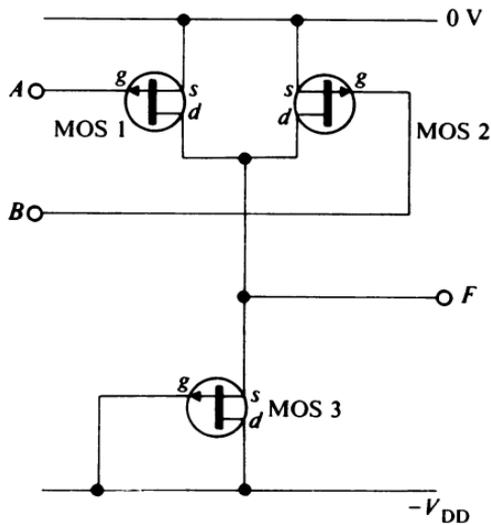


Fig. 20 Porte ET-NON MOS.

La figure 20 représente une porte élémentaire ET-NON (NAND) à transistors MOS à canal p . Lorsque l'une au moins des entrées A ou B est au niveau bas, le MOS correspondant est saturé, et la sortie F est au niveau haut (0 V). Lorsque les deux entrées A et B sont au niveau haut, le MOS 1 et le MOS 2 sont bloqués et la sortie F est au niveau bas ($-V_{DD}$). Cette porte réalise donc la fonction logique ET-NON (NAND).

Logique à MOS complémentaires (CMOS)

Cette famille logique utilise à la fois des transistors MOS à canal p et à canal n , montés symétriquement. Une porte élémentaire OU-NON (NOR) en CMOS est indiquée à la figure 21. Lorsque les deux signaux d'entrée A et B sont au niveau bas (0 V en logique positive), les transistors MOS 3 et MOS 4 sont bloqués, les transistors MOS 1 et MOS 2 sont passants, et la sortie F est au niveau haut ($+V_{DD}$ en logique positive). Lorsque l'entrée A est au niveau bas et que l'entrée B est au niveau haut, les transistors MOS 2 et MOS 4 sont bloqués, MOS 1 et MOS 3 sont passants, et la sortie F est au niveau bas. Il en est de même lorsque le signal d'entrée A est au niveau haut et que le signal d'entrée B est au niveau bas. Lorsque les deux signaux d'entrée A et B sont au niveau haut, les transistors MOS 1 et MOS 2 sont bloqués, les transistors MOS 3 et MOS 4 sont passants et la sortie F est encore au niveau bas.

Les niveaux haut et bas sont proches respectivement de $+V_{DD}$ et 0 V . Cela revient à dire que les circuits CMOS ont une très grande

immunité au bruit, de l'ordre de 20 % de la tension d'alimentation. Cette famille peut fonctionner avec des tensions d'alimentation comprises entre 3 et 15 V, mais pour des applications particulières comme les montres électroniques, la consommation ne dépasse guère 5 μA sous 1 V à 1,5 V, et un grand soin doit être apporté à la réduction des tensions de seuil.

L'impédance d'entrée d'un circuit MOS est en moyenne de $10^{12} \Omega$, avec une capacité de 5 pF. Le temps de propagation est d'environ 20 ns pour deux entrées; il augmente de 5 ns pour chaque augmentation de 5 pF de la charge. Si la vitesse n'est pas le plus important, la sortance peut aller jusqu'à 50.

Il faut faire extrêmement attention lorsque l'on manipule des circuits CMOS, car ces circuits peuvent être détruits par les tensions statiques et les équipements incorrectement mis à la terre. Ces circuits sont souvent livrés avec leurs broches enfoncées dans de la mousse graphitée. Il est très important d'attendre le tout dernier moment pour enlever la mousse et de ne pas utiliser de plastique ou de nylon lors du soudage, car ces matériaux sont sources de tensions électrostatiques élevées, tout comme les vêtements modernes en fibres synthétiques. De plus, tous les équipements doivent être reliés à la terre. La manipulation des circuits CMOS a créé de nouveaux problèmes, dont la plupart peuvent trouver des solutions satisfaisantes. Aux postes de câblage de ces circuits, une plaque en cuivre reliée à la terre a été ajoutée sur les établis et tous les appareils ont été reliés à la terre, ainsi que la personne travaillant sur ce poste, par l'intermédiaire de bracelets métalliques. Toutefois, la plupart des circuits intégrés MOS disponibles actuel-

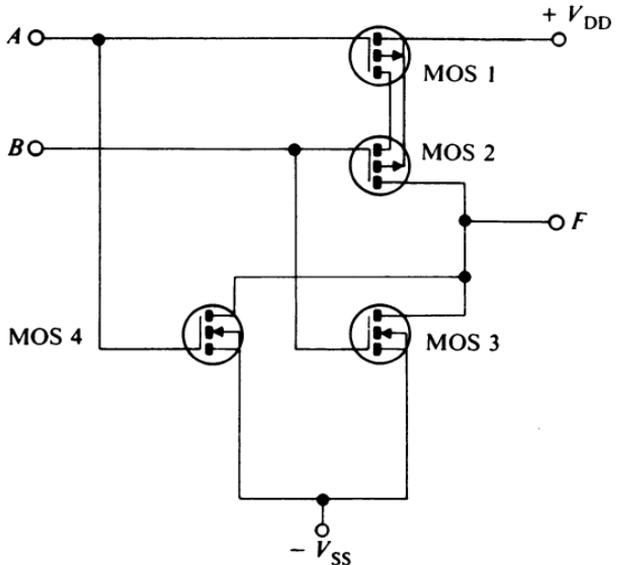


Fig. 21 Porte OU-NON CMOS.

lement sont protégés intérieurement par des étages tampons sur les entrées, mais il convient malgré tout de raccorder en premier lieu les broches d'alimentation.

REMARQUES :

- ne jamais laisser une entrée en l'air,
- raccorder les entrées inutilisées aux entrées employées, ou à l'alimentation au travers de résistances de 220 k Ω ,
- ne pas appliquer les signaux d'entrée tant que l'alimentation n'est pas raccordée et branchée.

Une grande partie des déboires connus avec les circuits CMOS provient de l'inobservation de ces quelques précautions élémentaires.

Les principaux avantages des CMOS résident dans leur forte densité d'intégration, leur gamme étendue de tension d'alimentation, leur forte immunité au bruit et leur faible consommation. Le fonctionnement sous 5 volts d'alimentation donne un temps de propagation de 35 ns, une consommation de 10 nW et une immunité au bruit de 2 V. Ces facteurs, ajoutés au faible coût de production ont conduit à une très grande utilisation des circuits intégrés CMOS pour les applications tant numériques que linéaires. Toutefois, les circuits et les applications décrits dans cet ouvrage s'appuient sur la technologie TTL bipolaire et les circuits de la série 74, quoique tout ce qui est logique est transposable aux circuits CMOS ou à tout autre circuit.

toute la vérité construire les logigrammes

Portes ET-NON (NAND) et OU-NON (NOR)

En pratique, une série déterminée de circuits intégrés ne comprend qu'un nombre limité de différents types de portes. En général, seules les portes ET-NON et OU-NON sont utilisées.

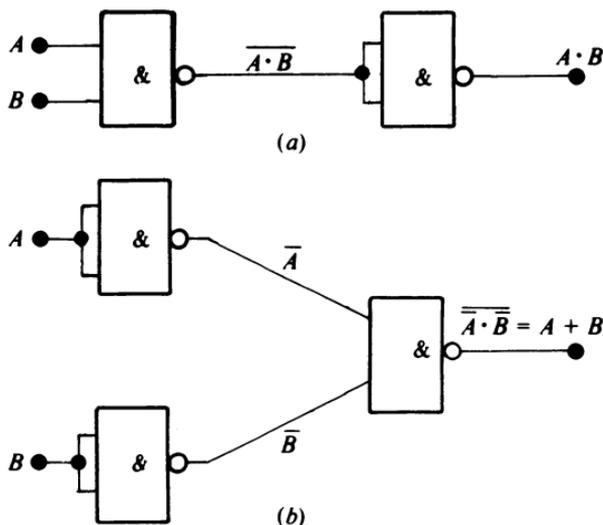


Fig. 1 Fonctions logiques de base réalisées avec des portes ET-NON :

- (a) fonction ET,
(b) fonction OU.

Les fonctions logiques de base à réaliser sont le ET, le OU et le NON. Il est possible de réaliser toutes ces fonctions en n'employant que des portes ET-NON (NAND) ou OU-NON (NOR). Les figures 1 *a* et 1 *b* montrent comment on peut réaliser les trois fonctions de base en n'employant que des portes ET-NON (NAND). Les figures 2 *a* et 2 *b* montrent comment on peut réaliser également les trois fonctions de base en n'employant que des portes OU-NON (NOR).

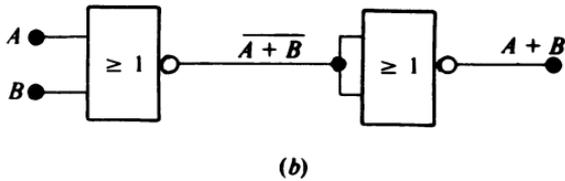
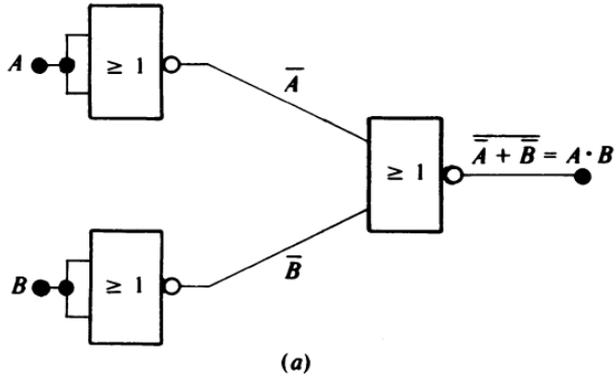


Fig. 2 Fonctions logiques de base réalisées avec des portes OU-NON :

- (a) fonction ET,
- (b) fonction OU.

Il est donc possible de construire un logigramme à partir des combinaisons précédentes. Il reste toutefois, un pas important à franchir : éliminer du logigramme les portes redondantes; c'est ce qu'on appelle la *minimisation*.

EXEMPLE 1

Construire un logigramme n'utilisant que des portes ET-NON pour réaliser la fonction F suivante :

$$F = A.B + C.D.$$

Le logigramme répondant à la fonction F avec des portes logiques de base est indiqué à la figure 3 *a*.

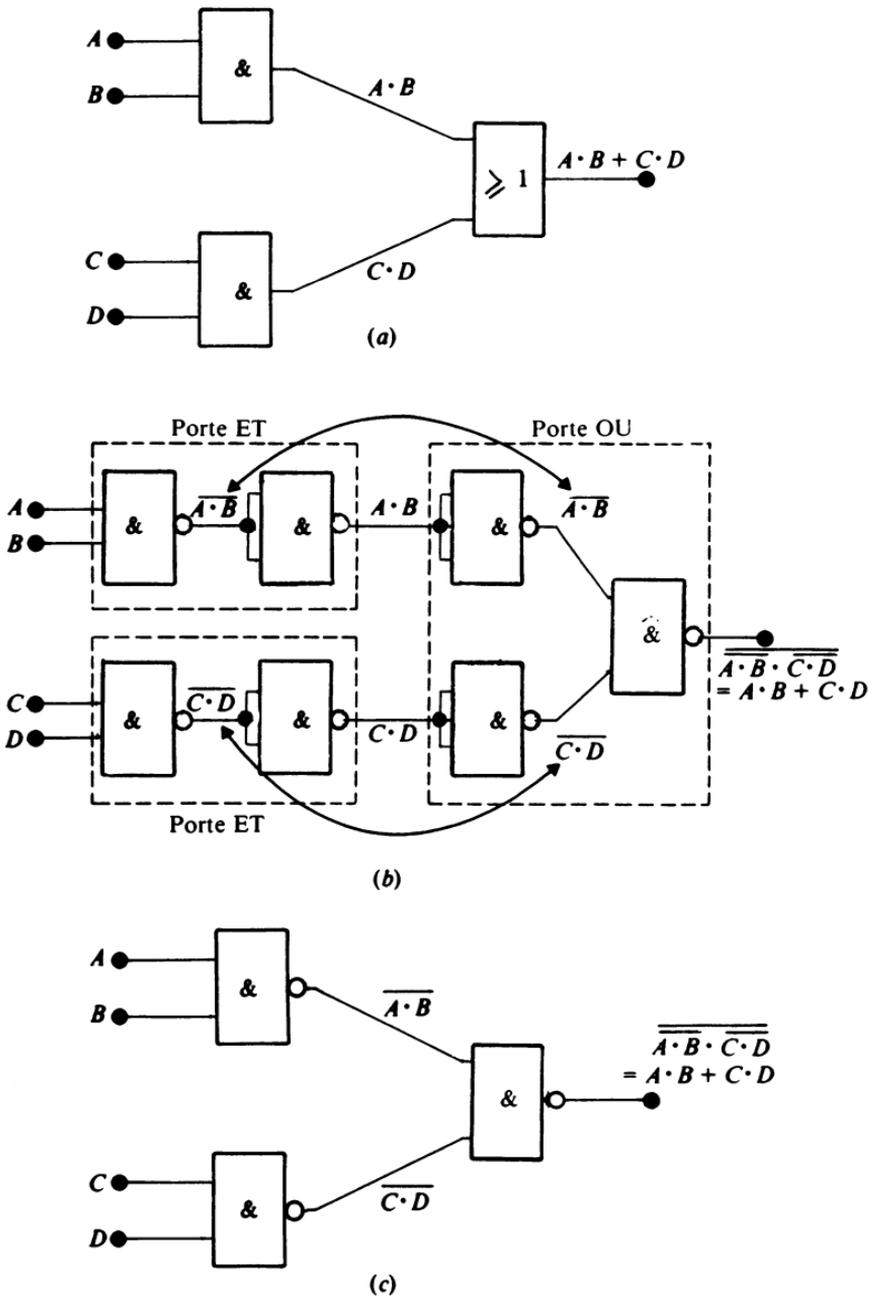


Fig. 3 Réalisation d'une fonction avec des portes ET-NON, et minimisation :
 (a) logigramme réalisé avec des portes logiques de base,
 (b) logigramme réalisé avec des portes ET-NON,
 (c) logigramme final après minimisation.

Si l'on remplace chaque porte logique de base par son équivalent en portes ET-NON (fig. 1), le nouveau logigramme devient celui de la figure 3 b.

Si maintenant, ce logigramme est examiné de près, on décèle l'existence de portes redondantes que l'on peut supprimer. Le logigramme final est donné à la figure 3 c.

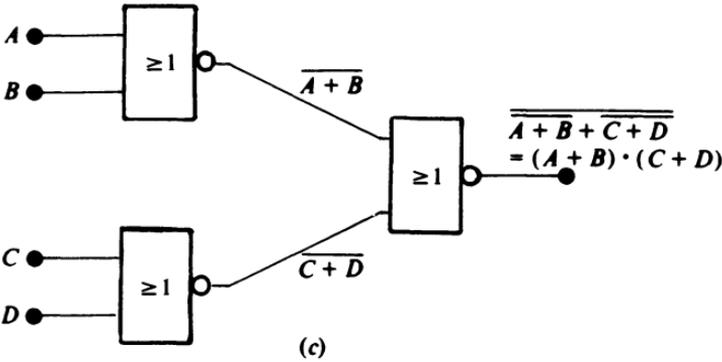
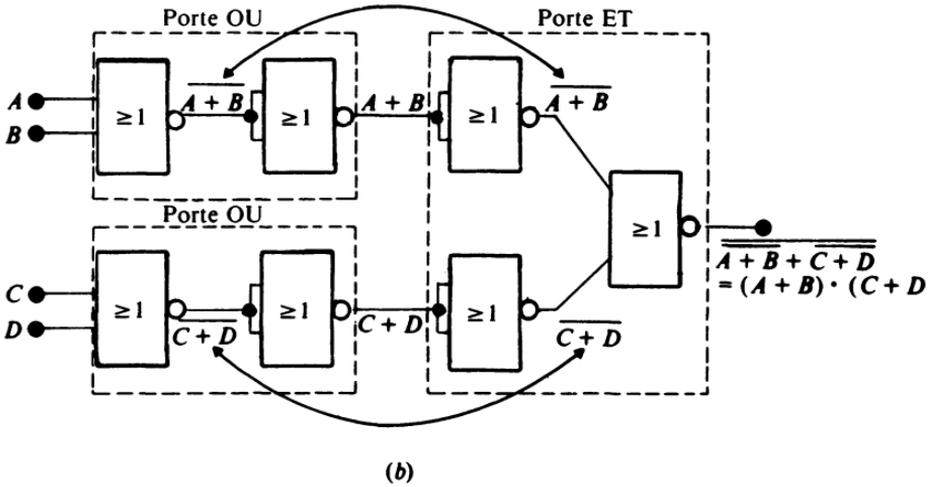
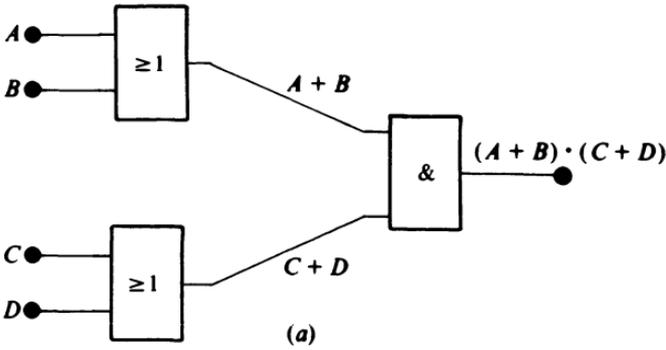


Fig. 4 Réalisation d'une fonction avec des portes OU-NON et minimisation :

- (a) logigramme réalisé avec les portes logiques de base,
- (b) logigramme réalisé avec des portes OU-NON,
- (c) logigramme final après minimisation.

EXEMPLE 2

Construire un logigramme n'utilisant que des portes OU-NON pour réaliser la fonction F suivante :

$$F = (A + B).(C + D)$$

Le logigramme répondant à la fonction F avec des portes logiques de base est indiqué à la figure 4 a.

Si l'on remplace chaque porte logique de base par son équivalent en portes OU-NON (fig. 2), le nouveau logigramme est indiqué à la figure 4 b.

A l'examen détaillé de ce logigramme, on décèle l'existence de portes redondantes que l'on peut supprimer. Le logigramme final est donné à la figure 4 c.

Porte OU exclusif

La porte OU exclusif est une porte très utile dans les systèmes logiques, car elle fournit un « 1 » en sortie lorsque les entrées sont différentes, c'est-à-dire $A = 0, B = 1$, ou $A = 1, B = 0$. Cette porte s'appelle également *porte d'anticoïncidence*, ou encore *comparateur*, car elle compare deux signaux logiques.

La fonction OU exclusif peut être décrite par l'équation booléenne :

$$F = A\bar{B} + \bar{A}B$$

Le logigramme de la figure 5 réalise cette fonction avec des portes logiques de base.

Le signal de sortie en X est égal à « 1 » lorsque $A > B$, A plus grand que B , c'est-à-dire lorsque $A = 1$ et $B = 0$, et le signal Y est égal à « 1 » lorsque $A < B$ (A plus petit que B), c'est-à-dire lorsque $A = 0$ et $B = 1$

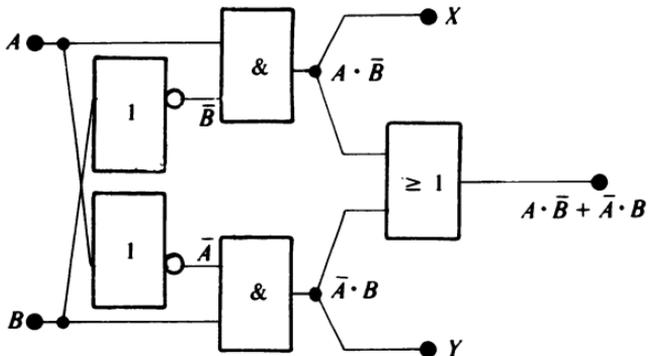


Fig. 5 Logigramme d'un OU exclusif réalisé avec des portes logiques de base.

La fonction OU exclusif peut être réalisée avec des portes ET-NON (NAND), tel que l'indique la figure 6.

REMARQUE : Il est conseillé de vérifier l'application des règles d'algèbre de Boole dans les logigrammes, de façon à se familiariser avec les différentes manipulations que l'on peut faire sur cet algèbre.

MONTAGE PRATIQUE

OU exclusif

- Réaliser les deux montages des figures 6 a et 6 b, en utilisant 1 1/4 ou 1 circuit intégré (quadruple porte ET-NON à 2 entrées) type SN 7400.
- Appliquer toutes les combinaisons possibles de signaux aux entrées et observer le niveau logique de sortie avec un indicateur à LED. Dresser la table de vérité de cette fonction logique.

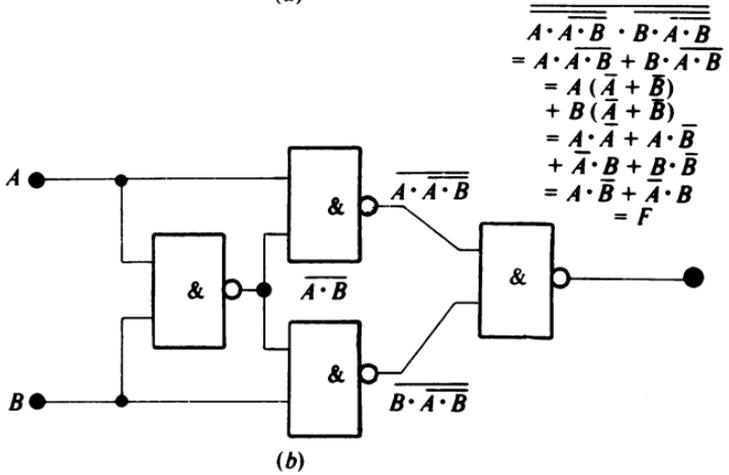
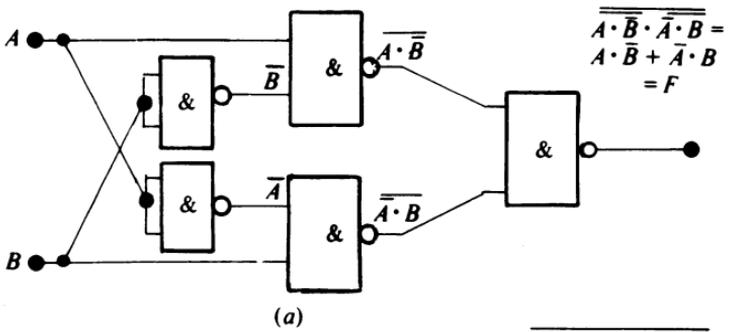


Fig. 6 Logigramme d'un OU exclusif réalisé avec portes ET-NON :

- (a) utilisant 5 portes ET-NON,
- (b) utilisant 4 portes ET-NON.

Comparteur

- Réaliser le montage de la figure 7, en utilisant les 3/4 d'un circuit SN 7400 (quadruple porte ET-NON à 2 entrées), et 1/3 d'un SN 7404 (sextuple inverseur).

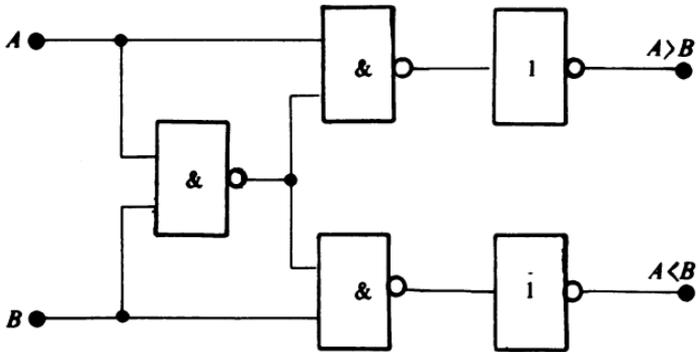


Fig. 7 Logigramme d'un comparateur.

- Appliquer toutes les combinaisons possibles de signaux aux entrées, et observer le niveau logique des deux sorties avec un indicateur à LED. Dresser la table de vérité de cette fonction logique.

Sécurité de machine

On considère une perceuse actionnée par un moteur électrique. Le moteur *F* ne peut fonctionner que si l'interrupteur de commande *S* est actionné et si les conditions suivantes de sécurité sont respectées :

- la protection de sécurité *G* est en place,
- le courant de surcharge *L* n'est pas dépassé.

Outre ces conditions normales de fonctionnement, une clé *K* doit permettre de faire tourner le moteur sans que la protection de sécurité soit en place.

En supposant que chaque variable *F*, *S*, *G* ou *L* vaut « 1 » lorsque la condition de sécurité est respectée, on aboutit alors à la table de vérité de la figure 8.

Sur la table de vérité il apparaît que la fonction *F* est réalisée (état logique 1) pour trois combinaisons différentes des variables. On peut donc écrire directement l'équation booléenne de la fonction *F* à partir de la table de vérité :

$$F = S.\bar{G}.L.K. + S.G.L.\bar{K}. + S.G.L.K$$

<i>S</i>	<i>G</i>	<i>L</i>	<i>K</i>	<i>F</i>
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Fig. 8 Table de vérité du système de sécurité.

Cette équation booléenne décrit la fonction logique, à partir des trois opérateurs de base. Le logigramme de la figure 9 montre comment on réalise cette fonction en se servant de ces trois opérateurs de base. Ce logigramme comporte 6 portes et 17 entrées, ce qui est un peu excessif.

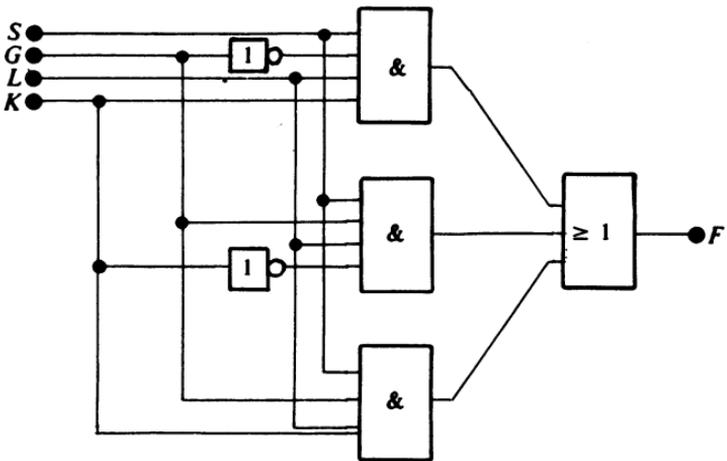


Fig. 9 Système de sécurité à portes logiques de base.

Si l'on ne dispose que de portes OU-NON (NOR) pour réaliser cette fonction, il faut mettre l'équation booléenne sous une forme convenant à ces portes, en utilisant les simplifications et les mises en facteur suivantes :

$$F = S.\bar{G}.L.K + S.G.L.\bar{K} + S.G.L.K$$

$$F = S.\bar{G}.L.K + S.G.L.(\bar{K} + K)$$

$$\text{mais } \bar{K} + K = 1$$

donc

$$F = S.\bar{G}.L.K + S.G.L = S.L.(\bar{G}.K + G)$$

$$\text{or } \bar{G}.K + G = (\bar{G} + G).(K + G) = K + G$$

$$\text{d'où : } F = S.L.(K + G)$$

Cette expression simplifiée est directement réalisable à l'aide de portes logiques de base. Il faut encore la transformer pour l'adapter aux portes OU-NON.

$$F = \overline{\bar{S}.L.(K + G)}$$

$$F = \overline{\bar{S} + \bar{L} + K + G}$$

$$F = \overline{\bar{S} + \bar{L} + K + G}$$

Cette expression correspond à un logigramme n'utilisant que des portes OU-NON, tel que l'indique la figure 10. Un tel logigramme n'utilise que 4 portes OU-NON (NOR) et 9 entrées.

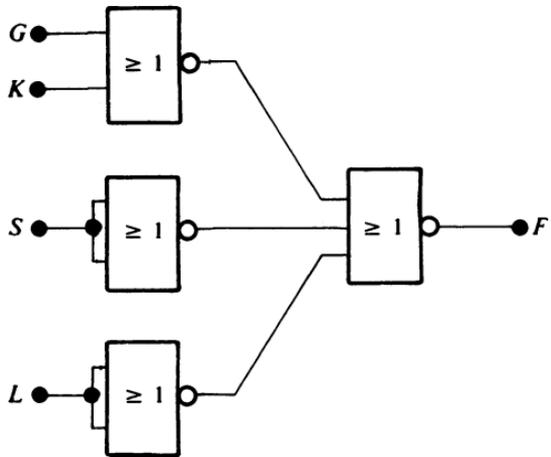


Fig. 10 Système de sécurité à portes OU-NON.

Autre supposition : on ne dispose que de portes ET-NON (NAND), pour réaliser cette fonction. Il faut donc mettre l'expression simplifiée précédente sous une forme convenant à des portes ET-NON.

$$F = S.L.(K + G)$$

$$\bar{F} = \bar{S} + \bar{L} + K.\bar{G}$$

$$F = S.L.(K.G)$$

$$F = S.L.(K.G)$$

Il est donc préférable de réaliser la fonction F à l'aide de porte ET-NON, puis à l'inverser par une porte ET-NON montée en inverseur, tel que l'indique la figure 11.

Un tel organigramme utilise 5 portes et 11 entrées.

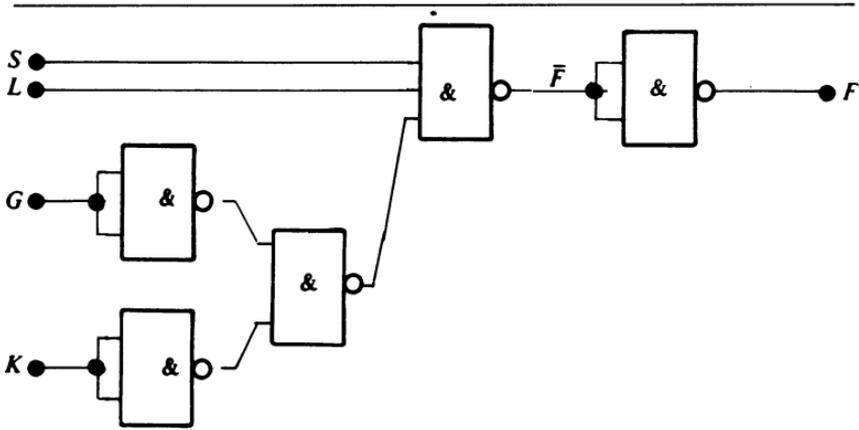


Fig. 11 Système de sécurité à portes ET-NON.

REMARQUES : Les méthodes de manipulation de l'algèbre de Boole, permettant de transformer une expression en une autre mieux adaptée à une réalisation à l'aide de portes OU-NON ou ET-NON, seront décrites plus en détail dans le chapitre suivant.

MONTAGE PRATIQUE

Sécurité de machine

- Réaliser le montage de la figure 10, en utilisant 3/4 de circuit SN 7402 (quadruple porte OU-NON à 2 entrées) et 1/2 circuit SN 7425 (double porte OU-NON à quatre entrées). Appliquer toutes les combinaisons possibles de signaux aux entrées, et vérifier la table de vérité de la figure 8.
- Réaliser également le montage de la figure 11, en utilisant 1 circuit SN 7400 (quadruple porte ET-NON à 2 entrées) et 1/2 circuit SN 7420 (double porte ET-NON à 4 entrées). Appliquer toutes les combinaisons possibles de signaux aux entrées, et vérifier la table de vérité de la figure 8.

les tableaux de Karnaugh

Le tableau de Karnaugh est une représentation graphique d'une table de vérité. Il se présente sous la forme d'un diagramme rectangulaire divisé en cases, dans lequel le nombre total de cases est égal à 2^N , avec N égal au nombre de variables.

\bar{A}	\bar{A}
A	A

(a)

	\bar{B}	B
\bar{A}	$\bar{A}.\bar{B}$	$\bar{A}.B$
A	$A.\bar{B}$	$A.B$

(b)

	\bar{B}	\bar{B}	B	B
\bar{A}	$\bar{A}.\bar{B}.\bar{C}$	$\bar{A}.\bar{B}.C$	$\bar{A}.B.C$	$\bar{A}.B.\bar{C}$
A	$A.\bar{B}.\bar{C}$	$A.\bar{B}.C$	$A.B.C$	$A.B.\bar{C}$
	\bar{C}	C	C	\bar{C}

(c)

Fig. 12 Tableaux de Karnaugh :

- (a) à une variable (2^1 cases),
- (b) à deux variables (2^2 cases),
- (c) à trois variables (2^3 cases).

Chaque variable représentée dans un tableau de Karnaugh utilise la moitié des cases, et son complément l'autre moitié.

Les figures 12 a, 12 b et 12 c montrent des tableaux de Karnaugh à une, deux ou trois variables. A chaque case correspond une combinaison des variables, un peu comme dans le jeu de la bataille navale.

La figure 13 a montre un tableau de Karnaugh à quatre variables. A l'intérieur de chaque case figure une des seize combinaisons possibles des quatre variables.

	\bar{C}	\bar{C}	C	C	
\bar{A}	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}\bar{B}\bar{C}D$	$\bar{A}\bar{B}C\bar{D}$	$\bar{A}\bar{B}CD$	\bar{B}
\bar{A}	$\bar{A}B\bar{C}\bar{D}$	$\bar{A}B\bar{C}D$	$\bar{A}BC\bar{D}$	$\bar{A}BCD$	B
A	$AB\bar{C}\bar{D}$	$AB\bar{C}D$	$ABC\bar{D}$	$ABCD$	B
A	$A\bar{B}\bar{C}\bar{D}$	$A\bar{B}\bar{C}D$	$A\bar{B}C\bar{D}$	$A\bar{B}CD$	\bar{B}
	\bar{D}	D	D	\bar{D}	

(a)

	CD			
AB	00	01	11	10
00	0000	0001	0011	0010
01	0100	0101	0111	0110
11	1100	1101	1111	1110
10	1000	1001	1011	1010

(b)

Fig. 13 Tableaux de Karnaugh et diagramme de Veitch à quatre variables :

(a) tableau de Karnaugh,

(b) diagramme de Veitch.

Si le niveau logique 1 est utilisé pour représenter chaque variable A , B , C , ou D , et le niveau logique « 0 » pour représenter leur complément \bar{A} , \bar{B} , \bar{C} , ou \bar{D} , on obtient un diagramme de Veitch, représenté à la figure 13 b. Toutefois, afin d'éviter toute confusion ultérieure, on appellera « Tableaux de Karnaugh » l'ensemble des deux représentations, car elles ne diffèrent que par la méthode de repérage des cases.

Il est bon de remarquer que deux cases adjacentes d'un tableau de Karnaugh ne diffère que par l'état d'une variable ou d'un élément binaire (fig. 13 a et 13 b). Il faut donc bien se représenter un tableau de Karnaugh comme un cylindre horizontal ou vertical,

dans lequel les cases du haut sont adjacentes aux cases du bas et les cases de droite adjacentes aux cases de gauche. Ceci est très important pour la suite.

Représentation graphique des fonctions

La représentation graphique des fonctions est une méthode rapide permettant de vérifier les théorèmes d'algèbre de Boole, de concevoir les logigrammes et de minimiser le nombre de portes logiques d'un circuit.

Regroupement des cases

Pour bien comprendre ce principe de regroupement des cases, il vaut mieux prendre des exemples :

EXEMPLE 1

Soit une fonction logique particulière, décrite par l'équation Booléenne suivante :

$$F = A.B.C.D + \bar{A}.B.C.D + \bar{A}.B.\bar{C}.D. + \bar{A}.\bar{B}.\bar{C}.D.$$

La table de vérité correspondante est indiquée à la figure 14 dans laquelle la fonction F vaut 1 pour chaque terme de l'équation booléenne, et vaut 0 pour les autres combinaisons des variables.

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>F</i>
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Fig. 14 Table de vérité de l'exemple 1.

On peut remplir les cases du tableau de Karnaugh, en partant de l'équation booléenne ou de la table de vérité. Il faut noter que l'on doit placer la valeur logique (0 ou 1) de la fonction dans chaque case correspondante du tableau. Il y a donc quatre cases qui valent 1 et les 12 cases restantes qui valent 0 (fig. 15 a).

Colonnes

		1	2	3	4	
				C		
Lignes	1	0	1	0	0	}
	2	0	1	1	0	
	3	0	0	1	0	
	4	0	0	0	0	
		D				

(a)

Colonnes

		1	2	3	4	
				C		
Lignes	1	0	1	0	0	}
	2	0	1	1	0	
	3	0	0	1	0	
	4	0	0	0	0	
		D				

(b)

Fig. 15 Tableaux de Karnaugh de l'exemple 1 :

(a) remplissage du tableau de Karnaugh,
 (b) groupement de cases.

Les cases adjacentes des lignes 1 et 2 de la colonne 2, correspondant aux termes $\bar{A}.\bar{B}.\bar{C}.D$ et $\bar{A}.B.\bar{C}.D$ de l'équation booléenne précédente. Ces deux termes se simplifient de la façon suivante :

$$\bar{A}.\bar{B}.\bar{C}.D + \bar{A}.B.\bar{C}.D = \bar{A}.\bar{C}.D.(B + \bar{B}) = \bar{A}.\bar{C}.D$$

De façon identique, les cases adjacentes des lignes 2 et 3 de la colonne 3 correspondent aux termes $\bar{A}.B.C.D$ et $A.B.C.D$ de

l'équation booléenne précédente. Ces deux termes se simplifient : $\bar{A}.B.C.D + A.B.C.D = B.C.D.(\bar{A} + A) = B.C.D$

Ainsi, l'équation booléenne simplifiée peut s'écrire :

$$F = \bar{A}.\bar{C}.D + B.C.D.$$

Le tableau de Karnaugh de la figure 15 a, montre que les possibilités de simplification s'obtiennent en regroupant les cases, tel que l'indique la figure 15 b.

REMARQUE : Il est possible de regrouper les cases par puissances de deux, c'est-à-dire par 2, par 4, par 8, etc. Il est recommandé de chercher à regrouper le plus grand nombre possible de cases, car les simplifications obtenues sont plus importantes. Les regroupements peuvent s'entrecroiser et se superposer.

Lorsque l'on regroupe *deux* cases, on ramène à un seul terme les deux termes correspondants de l'équation booléenne. La variable qui change de valeur dans les deux cases n'y figure plus.

Lorsque l'on regroupe *quatre* cases, on ramène à un seul terme les quatre termes correspondants de l'équation booléenne. Les deux variables qui changent de valeur dans les quatre cases n'y figurent plus.

Lorsque l'on regroupe *huit* cases, on ramène à un seul terme les huit termes correspondants de l'équation booléenne. Les trois variables qui changent de valeur dans les huit cases disparaissent de ce terme.

De ce fait, dans l'exemple précédent, le simple examen du tableau de Karnaugh (fig. 15 b) montre que la fonction simplifiée ne comprend que deux termes, puisque l'on effectue deux boucles.

$$F = \bar{A}.\bar{C}.D + B.C.D$$

EXEMPLE 2

Sur le tableau de Karnaugh de la fonction F suivante :

$$F = A.\bar{B}.\bar{C} + A.B.\bar{C}$$

La méthode de regroupement des cases est utilisée pour effectuer une simplification. Cela donne le tableau de la figure 16.

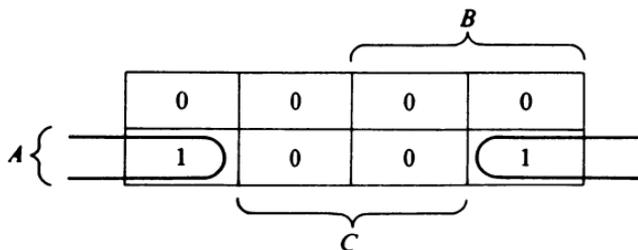


Fig. 16 Tableau de Karnaugh de l'exemple 2.

De la seule boucle y figurant, on en déduit :

$$F = A.\bar{C}$$

EXEMPLE 3

Soit la fonction F suivante :

$$F = A.\bar{B}.\bar{C}.D + A.\bar{B}.C.\bar{D} + \bar{A}.B.C.D + \bar{A}.B.\bar{C}.D$$

Pour simplifier cette fonction la méthode de regroupement des cases est appliquée au tableau de Karnaugh correspondant. Cela donne le tableau de la figure 17.

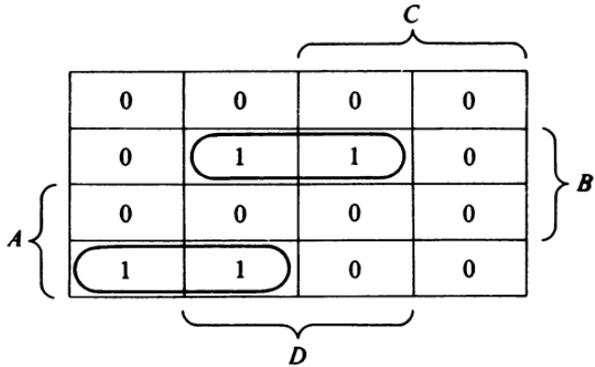


Fig. 17 Tableau de Karnaugh de l'exemple 3.

Deux boucles y figurent; on en déduit

$$F = A.\bar{B}.\bar{C} + \bar{A}.B.D.$$

EXEMPLE 4

Soit la fonction F décrite par l'équation booléenne suivante :

$$F = A.\bar{B}.\bar{C}.\bar{D} + A.\bar{B}.C.\bar{D} + \bar{A}.B.C.D + \bar{A}.B.D.C$$

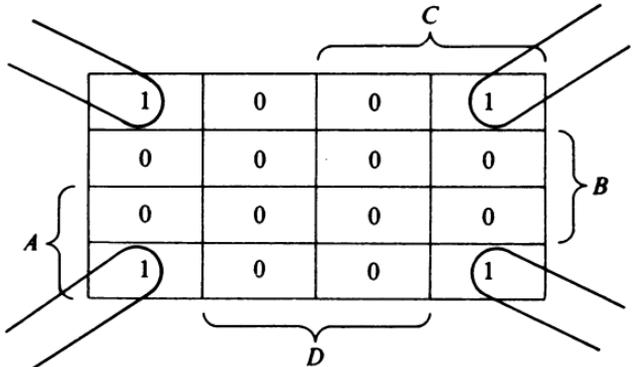


Fig. 18 Tableau de Karnaugh de l'exemple 4.

Le tableau de Karnaugh simplifié (regroupement des cases) est donné en figure 18.

On déduit de la seule boucle qui y apparaît :

$$F = \bar{B}.\bar{D}$$

MONTAGE PRATIQUE

Écrire les tableaux de Karnaugh des fonctions F décrites par les équations booléennes suivantes :

$$F_1 = A.\bar{B}.C.D + A.\bar{B}.\bar{C}.D + A.B.C.D + A.B.\bar{C}.D$$

$$F_2 = A.B.\bar{C}.\bar{D} + \bar{A}.B.\bar{C}.\bar{D} + \bar{A}.B.C.\bar{D} + A.B.C.\bar{D}$$

$$F_3 = \bar{A}.\bar{B}.\bar{C}.D + \bar{A}.\bar{B}.C.D + A.\bar{B}.\bar{C}.D + A.\bar{B}.C.D$$

$$F_4 = \bar{A}.\bar{B}.\bar{C}.\bar{D} + \bar{A}.\bar{B}.C.\bar{D}$$

Ces tableaux de Karnaugh permettent donc de résumer sous une forme compacte les tables de vérité. Les exemples suivants vont indiquer la manière d'utiliser les tableaux de Karnaugh pour obtenir une expression directement réalisable avec des portes logiques ET-NON (NAND), OU-NON (NOR).

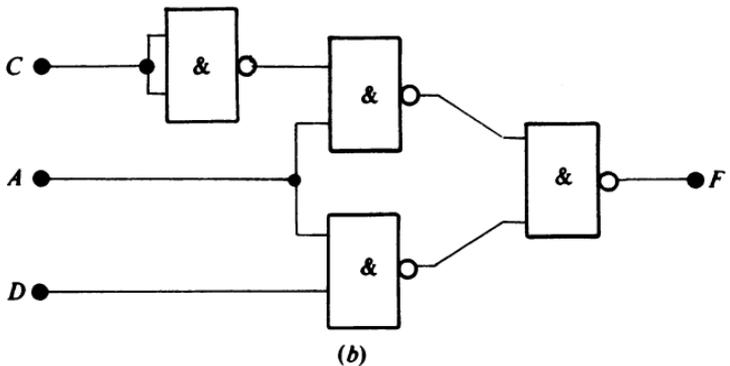
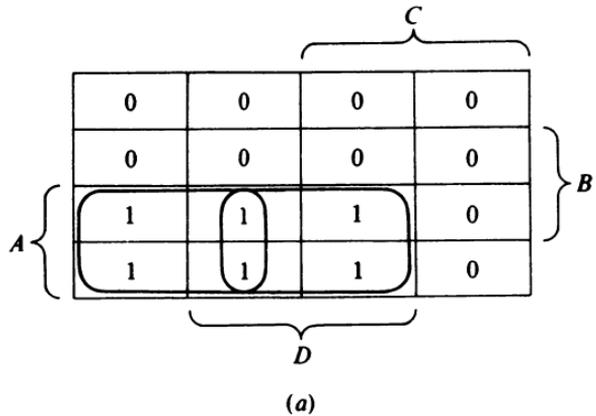


Fig. 19 Tableau de Karnaugh et logigramme de l'exemple 5 :

(a) tableau de Karnaugh,

(b) logigramme n'utilisant que des portes ET-NON.

EXEMPLE 5

Soit le tableau de Karnaugh de la figure 19 a, décrivant la fonction logique F , dont le logigramme de réalisation ne comprend que des portes ET-NON.

Des deux boucles qui y figurent, il s'en déduit la fonction F simplifiée :

$$F = A.D + A.\bar{C}$$

Il faut modifier cette expression, de façon à ce qu'elle s'adapte à une réalisation n'utilisant que des portes ET-NON :

$$\bar{F} = \overline{A.D + A.\bar{C}} = \overline{A.D} . \overline{A.\bar{C}}$$

$$\text{soit } F = \overline{\overline{A.D} . \overline{A.\bar{C}}}$$

Cette expression ne comprend plus que des termes pouvant être réalisés directement par des portes ET-NON (logigramme de la figure 19 b).

EXEMPLE 6

On considère la même fonction logique de l'exemple 5 et son tableau de Karnaugh correspondant (fig. 20 a).

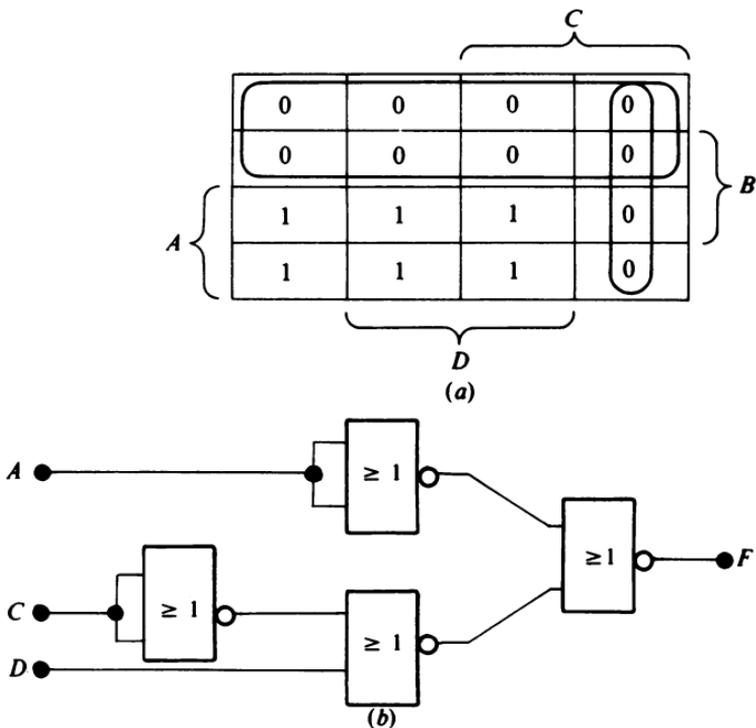


Fig. 20 Tableau de Karnaugh et logigramme de l'exemple 6 :

(a) tableau de Karnaugh,

(b) logigramme n'utilisant que des portes OU-NON.

Mais le logigramme de réalisation de cette fonction ne comprend cette fois que des portes OU-NON.

Il est préférable d'effectuer des regroupements de « 0 » dans le tableau de Karnaugh, lorsque l'on désire ne réaliser le logigramme qu'avec des portes OU-NON. On obtient ainsi la fonction complémentaire F , qu'il suffit alors d'inverser.

Des deux boucles figurant dans le tableau de Karnaugh (fig. 20 a), se déduit le complément de la fonction F :

$$\bar{F} = \bar{A} + C\bar{D},$$

réécrite pour l'adapter à une réalisation à des portes OU-NON :

$$\bar{F} = \bar{A} + \overline{C + D}, \text{ soit } F = \overline{\bar{A} + \overline{C + D}}.$$

Le logigramme correspondant est indiqué en figure 20 b.

EXEMPLE 7

Dans un système logique à quatre variables d'entrée désignées par A , B , C et D , les variables A et B représentent un nombre binaire X dont A est l'élément binaire de poids fort, et les variables C et D un nombre binaire Y donc C est l'élément binaire de poids fort.

X		Y		F
A	B	C	D	
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Fig. 21 Table de vérité de l'exemple 7.

La sortie F vaut « 1 » lorsque X est plus grand que Y . La sortie F vaut « 0 » pour toutes les autres combinaisons des variables. La table de vérité de ce système logique est donnée en figure 21. Cela conduit au tableau de Karnaugh de la figure 22 *a*, avec simplification de l'équation de F par regroupement des cases. Comme l'on veut réaliser un logigramme à portes ET-NON les cases contenant les « 1 » sont regroupées.

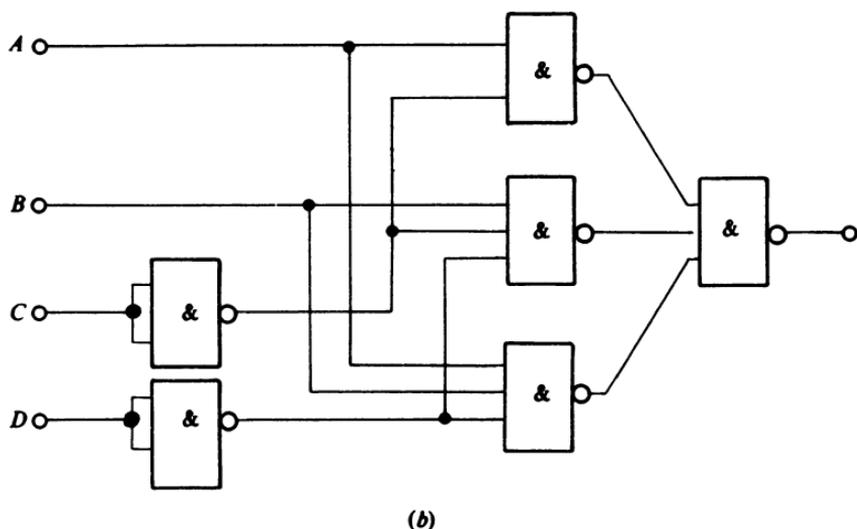
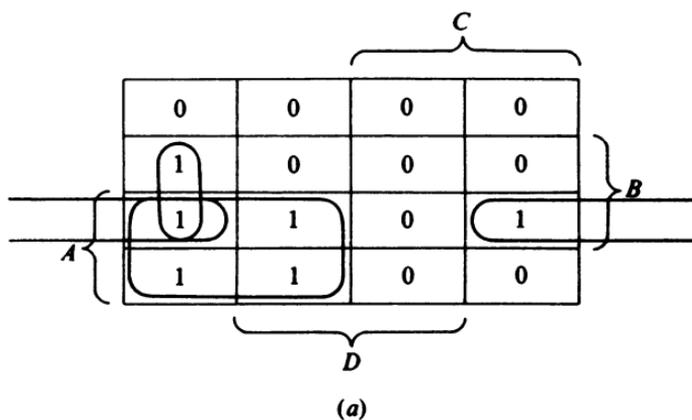


Fig. 22 Tableau de Karnaugh et logigramme de l'exemple 7 :

(a) tableau de Karnaugh,

(b) logigramme n'utilisant que des portes ET-NON.

Des trois boucles réalisées dans le tableau de Karnaugh, se déduit l'équation booléenne de la fonction F :

$$F = A.\bar{C} + B.\bar{C}.D + A.B.D$$

modifiée comme suit, pour l'adapter à une réalisation à portes ET-NON :

$$\overline{F} = \overline{A.C} + \overline{B.C.D} + \overline{A.B.D} = \overline{A.C} . \overline{B.C.D} . \overline{A.B.D}$$

et finalement $F = \overline{\overline{A.C} . \overline{B.C.D} . \overline{A.B.D}}$

A partir de cette équation, on peut construire le logigramme indiqué à la figure 22 b.

EXEMPLE 8

Soit un système logique similaire à celui de l'exemple 7, dans lequel F vaut « 1 » lorsque X est plus grand ou égal à Y (nombres binaires). La sortie F vaut « 0 » pour toutes les autres combinaisons des variables.

La table de vérité de ce système est indiquée en figure 23.

X		Y		F
A	B	C	D	
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Fig. 23 Table de vérité de l'exemple 8.

On obtient le tableau de Karnaugh correspondant à la figure 24 a. Pour réaliser un logigramme n'utilisant que des portes OU-NON, les cases contenant les « 0 » sont regroupées.

Trois boucles peuvent être faites dans le tableau de Karnaugh. L'équation booléenne simplifiée de la fonction F est alors :

$$\overline{F} + \overline{A}.\overline{B}.D + \overline{A}.C + \overline{B}.C.D$$

qu'il faut modifier pour une réalisation à portes OU-NON :

$$F = \overline{A + B + \overline{D}} + \overline{A + \overline{C}} + \overline{B + \overline{C} + \overline{D}}$$

$$\text{et finalement } F = \overline{A + B + \overline{D}} + \overline{A + \overline{C}} + \overline{B + \overline{C} + \overline{D}}$$

A partir de cette équation, on obtient le logigramme de la figure 24 b.

EXEMPLE 9

Les quatre variables d'entrée, A , B , C et D d'un système logique, représentent un nombre binaire à 4 éléments binaires (ou bits),

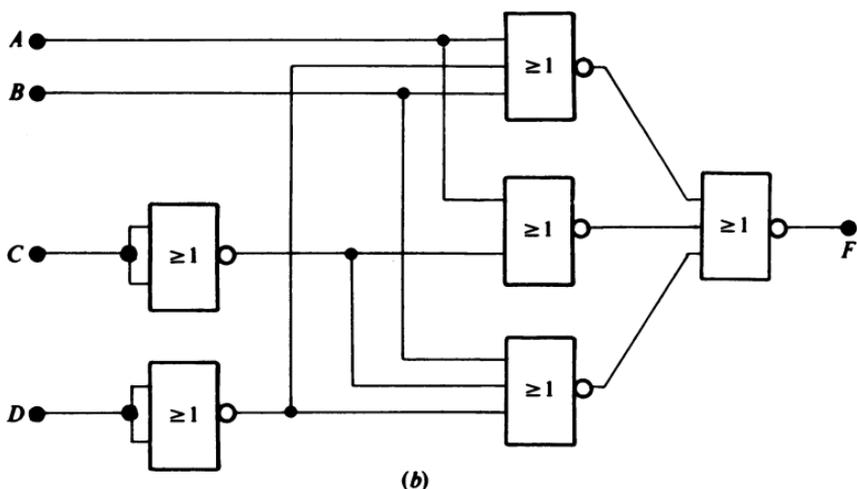
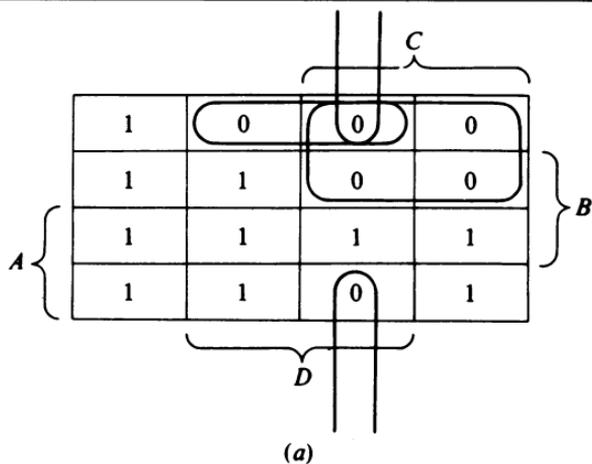


Fig. 24 Tableau de Karnaugh et logigramme de l'exemple 8 :

(a) tableau de Karnaugh,

(b) logigramme n'utilisant que des portes OU-NON.

dans lequel A est l'élément binaire de poids fort. Si le nombre binaire est inférieur ou égal au huit décimal, la fonction F vaut 1. Si le nombre est supérieur au huit décimal et s'il est inférieur ou égal au onze décimal, la fonction F vaut « 0 ». Si le nombre est supérieur au onze décimal, la fonction F peut valoir indifféremment « 0 » ou « 1 ».

Numéro décimal	A	B	C	D	F
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	*
13	1	1	0	1	*
14	1	1	1	0	*
15	1	1	1	1	*

Fig. 25 Table de vérité de l'exemple 9.

La table de vérité est donnée en figure 25; l'astérisque (*) désigne ici un état logique qui peut être indifféremment égal à 0 ou 1.

On peut, lors des regroupements de cases, sur le tableau de Karnaugh de ce système, décider de la valeur des cases optionnelles contenant un astérisque, afin de réaliser des regroupements plus grands. La figure 26 illustre bien ceci.

A partir des deux boucles réalisées dans le tableau de Karnaugh, se déduit l'équation booléenne de la fonction F :

$$F = \bar{A} + \bar{C}.D$$

A titre d'exercice, on peut réaliser les logigrammes de cette fonction, en n'utilisant que des portes ET-NON (NAND) ou des portes OU-NON (NOR).

		C		
	1	1	1	1
	1	1	1	1
A	*	*	*	*
	1	0	0	0
		D		

Fig. 26 Tableau de Karnaugh de l'exemple 9.

REMARQUE : Avec un peu d'expérience, il est possible de remplir directement le tableau de Karnaugh, sans passer par l'étape de la table de vérité, puisque ces deux éléments ne sont que deux formes différentes de représentation d'une fonction logique.

les bascules en balance

différents types de bistables

Les bascules sont des éléments bistables, c'est-à-dire des éléments ayant deux états stables de fonctionnement. Le passage d'un état à un autre est ordonné par un ou plusieurs signaux de commande. Les bascules sont des mémoires élémentaires, puisque l'état de la sortie indique l'état dans lequel était l'entrée, quelques instants auparavant. Les systèmes logiques employant des bascules font intervenir le temps : on les appelle donc des systèmes séquentiels. Les diverses bascules que l'on va examiner, diffèrent par le nombre et l'allure des signaux de commande dont elles ont besoin.

Bascule *RS*

Le symbole logique de la bascule *RS* est donné à la figure 1, ainsi que son logigramme constitué par deux portes OU-NON interconnectées, à l'aide duquel on peut simuler le fonctionnement de la bascule *RS*.

Par définition, l'application du niveau logique « 1 » sur l'entrée *S* (de l'anglais « set » qui veut dire « mise à un »), entraîne un état logique « 1 » sur la sortie *Q*, quel que soit l'état logique antérieur de cette sortie *Q*, et un état logique « 0 » sur la sortie \overline{Q} . L'application du niveau logique « 1 » sur l'entrée *R* (« reset » ou « remise à zéro »), entraîne un état logique « 0 » sur la sortie *Q*, quel que soit l'état logique antérieur de cette sortie *Q*, et un état logique « 1 » sur la sortie \overline{Q} .

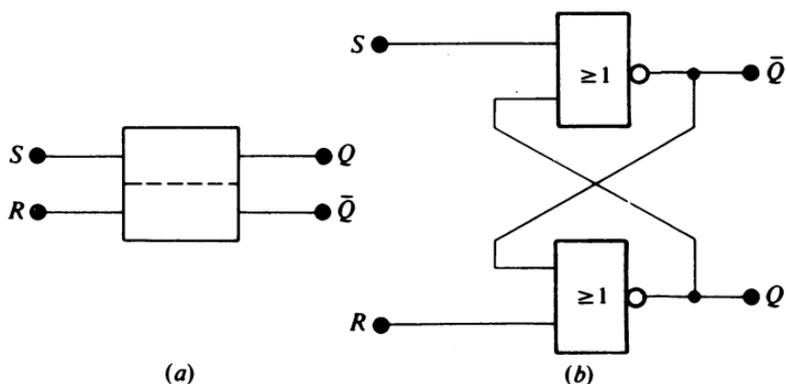


Fig. 1 Bascule RS :

(a) symbole logique,
(b) logigramme.

L'application simultanée d'un niveau logique « 1 » sur les entrées S et R entraîne un état indéterminé des sorties Q et \bar{Q} . Ceci doit être évité, car elle entraîne une non complémentarité des sorties Q et \bar{Q} , et des oscillations pouvant aller jusqu'à détruire le circuit. La table de vérité de cette bascule RS est donnée à la figure 2; Q_{t-1} représente l'état de la sortie Q avant l'application des signaux de commande sur R et sur S , et Q_t l'état de la sortie Q après l'application des signaux de commande sur R et sur S .

REMARQUE : Lorsque le logigramme de la bascule RS est constitué par deux portes OU-NON, l'application simultanée d'un niveau logique « 1 » sur les entrées S et R entraîne un état « 0 » bien défini des deux sorties Q et \bar{Q} .

S	R	Q_{t-1}	Q_t
0	0	0	0
0	0	1	1
1	0	0	1
1	0	1	1
0	1	0	0
0	1	1	0
1	1	0	*
1	1	1	*

} Etats indéterminés

Fig. 2 Table de vérité de la bascule RS.

Bascule RS

- Réaliser le montage de la figure 3.
- Appliquer toutes les combinaisons des signaux aux entrées *S* et *R* de façon à établir la table de vérité de ce montage.

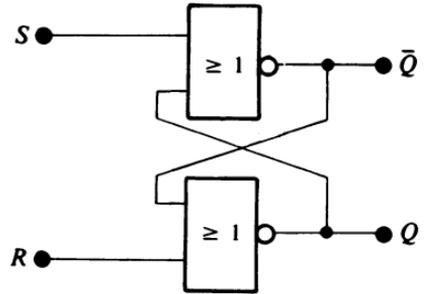


Fig. 3 Logigramme de la bascule RS.

- Refaire cet exercice en utilisant cette fois deux portes ET-NON à deux entrées, au lieu de portes OU-NON.

REMARQUE : Les signaux à appliquer sur les entrées *S* et *R* de cette bascule RS réalisée avec des portes ET-NON doivent être inversés par rapport à ceux d'une bascule RS réalisée avec des portes OU-NON.

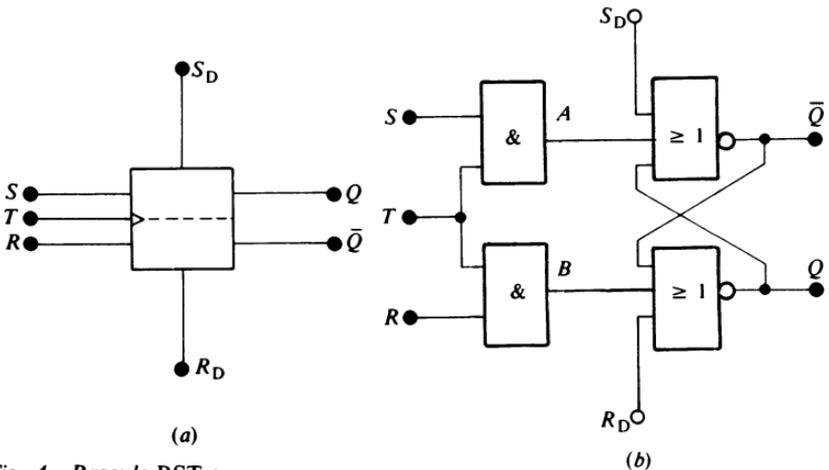


Fig. 4 Bascule RST :
 (a) symbole logique,
 (b) logigramme.

Bascule RST

Le symbole logique de la bascule *RST* est donné à la figure 4, ainsi que son logigramme ne différant de celui de la bascule *RS* que par la présence d'une entrée d'horloge *T*.

Lorsque cette entrée d'horloge est à « 1 », les points *A* et *B* sont reliés aux points *S* et *R*, permettant alors à la bascule de changer d'état.

Des entrées de forçage supplémentaires (S_D : mise à « 1 » directe, et R_D : remise à « 0 » directe) sont maintenant nécessaires pour forcer la bascule *RST* à prendre un état donné, quel que soit le niveau logique de l'entrée d'horloge.

MONTAGE PRATIQUE

Bascule RST

- Réaliser le montage de la figure 5. Il correspond au logigramme de la bascule *RST*.
- Régler la fréquence d'un signal rectangulaire produit par un générateur d'horloge type 555, à une valeur permettant de vérifier le fonctionnement du système, lorsque l'on applique des niveaux logiques aux entrées *S* et *R* de ce système.
- Débrancher le générateur d'horloge et vérifier l'influence des entrées de forçage S_D et R_D sur les niveaux logiques des sorties *Q* et \bar{Q} .

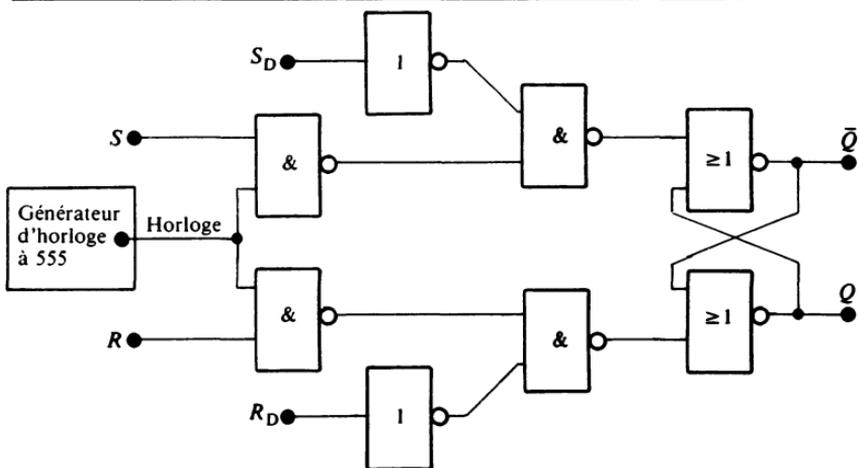


Fig. 5 Logigramme de la bascule *RST*.

Bascule *D*

La bascule *D* résout le problème de non complémentarité des sorties *Q* et \bar{Q} de la bascule *RST*, en rendant complémentaire en per-

manence les entrées R et S à l'aide d'un inverseur. Le logigramme d'une telle bascule est donné à la figure 6.

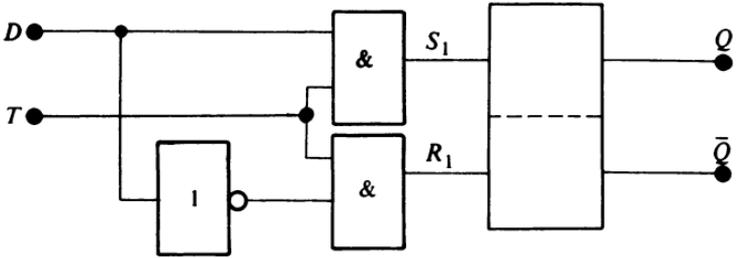


Fig. 6 Bascule D

La table de vérité de cette bascule D (fig. 7) montre que la sortie Q_t reproduit fidèlement les valeurs logiques de l'entrée D .

D	Q_{t-1}	Q_t
0	0	0
0	1	0
1	0	1
1	1	1

Fig. 7 Table de vérité de la bascule D

MONTAGE PRATIQUE

Bascule D

Réaliser le montage de la figure 8, correspondant à une bascule D et vérifier son fonctionnement.

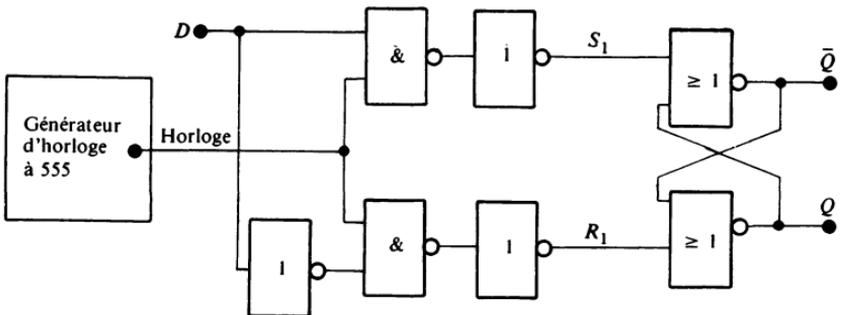


Fig. 8 Logigramme de la bascule D

Structure maître-esclave

Les bascules RST et D fonctionnent dès que le signal d'horloge devient haut, c'est-à-dire lorsqu'il est égal à 1 (fonctionnement sur front montant). De ce fait, lorsque ces bascules sont utilisées dans un compteur, il peut être gênant que les sorties des bascules changent d'état au moment où les niveaux d'entrée des bascules suivantes devraient être fixes, pour y être enregistrés. La structure maître-esclave a été développée pour s'affranchir de ce problème de temps.

La figure 9 donne la structure maître-esclave d'une bascule RST , comprenant deux bascules RST en série, dont les entrées d'horloge sont complémentaires.

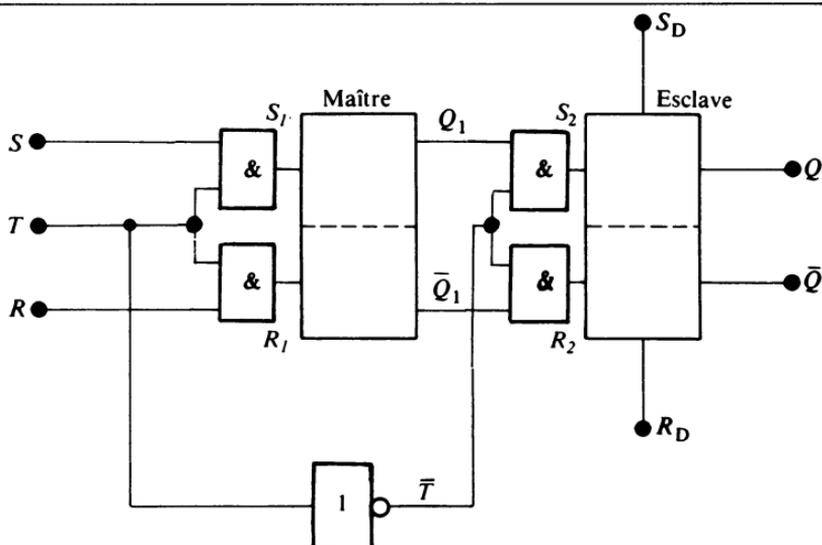


Fig. 9 Bascule RST maître-esclave.

La bascule RST maître fonctionne dès que le signal d'horloge passe à 1 (front montant). Pendant ce temps, la bascule RST esclave est bloquée, puisque son signal d'horloge est à « 0 ». Lorsque le signal d'horloge de la bascule maître passe à « 0 », la bascule esclave prend l'état de la bascule maître puisque le signal d'horloge de la bascule esclave est alors égal à « 1 ». Ainsi la bascule RST maître-esclave *enregistre* l'information dès le front montant de l'horloge, et elle *communique* cette information à des sorties Q et \bar{Q} dès le front descendant de l'horloge. On obtient ainsi un effet de *sas* propre à l'utilisation de ces bascules en comptage.

Bascule JK

La bascule JK est constituée de deux bascules RST associées en structure maître-esclave, avec une rétroaction croisée entre les sor-

ties de la bascule *RST* esclave et les entrées de la bascule *RST* maître, tel que l'indique la figure 10.

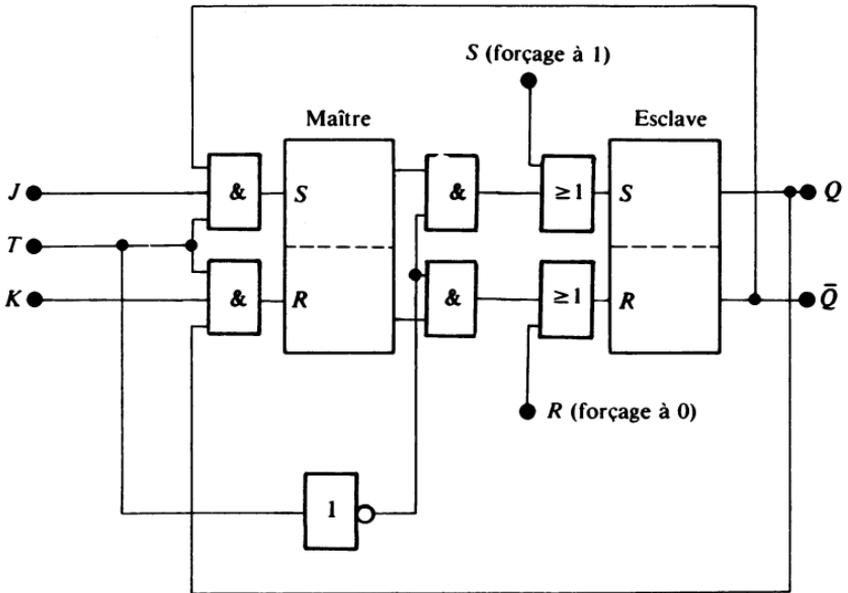


Fig. 10 Bascule JK.

La table de vérité est donnée à la figure 11. On constate que la sortie Q_t :

- ne change pas d'état, lorsque $J = 0$ et $K = 0$,
- est égale à 0, lorsque $J = 0$ et $K = 1$,
- est égale à 1, lorsque $J = 1$ et $K = 0$,
- change d'état, lorsque $J = 1$ et $K = 1$.

J	K	Q_{t-1}	Q_t
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Fig. 11 Table de vérité de la bascule JK.

Lorsque les signaux sur J et sur K sont égaux à 1, la bascule JK fonctionne alors en diviseur par deux, car la sortie Q change

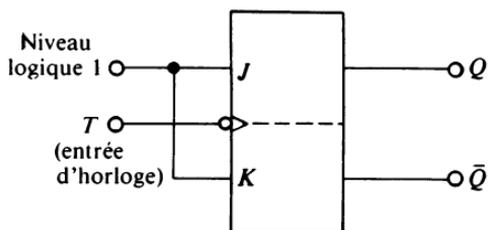


Fig. 12 Bascule JK câblée en diviseur par deux.

d'état pour deux changements d'état du signal d'horloge. La figure 12 montre une bascule *JK* montée en diviseur par deux. Il est également possible de réaliser un diviseur par deux avec une bascule *D* à autoblocage (non décrite dans cet ouvrage), tel que l'indique la figure 13.

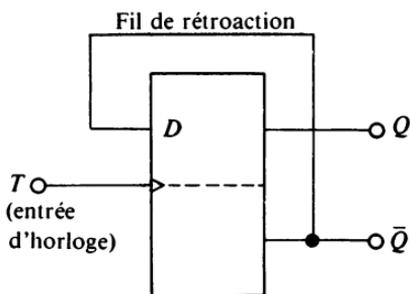


Fig. 13 Bascule *D* à autoblocage câblée en diviseur par deux.

Les bascules *JK* à structure maître-esclave et les bascules *D* à autoblocage sont les bascules les plus employées dans les systèmes logiques, et se présentent intégrées sous différentes configurations, tel que le circuit SN 7474 (double bascule *D* à autoblocage avec entrées séparées de mise à « 1 » et de remise à « 0 ») ou le circuit SN 74107 (double bascule *JK* avec entrées séparées d'horloge et de remise à « 0 »).

une lourde addition

l'arithmétique binaire

Les quatre opérations arithmétiques binaires, c'est-à-dire l'addition, la soustraction, la multiplication et la division, sont réalisable grâce à des manipulations sur l'addition. L'addition est donc une opération très importante, utilisant deux types de circuits : les demi-additionneurs et les additionneurs complets.

Un demi-additionneur fournit la *somme* et la *retenue* de deux chiffres binaires. Un additionneur complet fournit la *somme* et la *retenue* de deux chiffres binaire et de la *retenue précédente*. Les additionneurs peuvent fonctionner en mode série ou en mode parallèle. En mode série, l'addition des différents chiffres binaires s'effectue chiffre après chiffre, en commençant par le chiffre de plus faible poids. En mode parallèle, l'addition de tous les chiffres s'effectue simultanément. Les additionneurs parallèles sont plus rapides à effectuer l'opération d'addition que les additionneurs séries, mais en contrepartie, il sont plus complexes et plus chers.

L'addition binaire

La démarche mathématique est la même pour l'addition, quel que soit la base dans laquelle elle est utilisée.

En commençant par le chiffre de plus faible poids (à l'extrême droite), on additionne les « 1 » de la façon habituelle :

- si une colonne comprend un seul « 1 », le résultat est égal à 1 et on ne reporte aucune retenue dans la colonne suivante,

- si une colonne comprend deux « 1 », le résultat est égal à « 0 » et on reporte un « 1 » de retenue dans la colonne suivante,
- si une colonne comprend trois « 1 », dont un provient d'une retenue précédente, le résultat est égal à « 1 » et on reporte un « 1 » de retenue dans la colonne suivante.

EXEMPLE

Soit l'addition binaire des deux chiffres décimaux 46 et 22 :

$$\begin{array}{r}
 101110 = 46 \\
 + 10110 = 22 \\
 \hline
 1000100 = 68
 \end{array}$$

Le demi-additionneur est l'élément logique de base de l'opération d'addition. La figure 1 donne le logigramme d'un demi-additionneur, constitué d'un OU exclusif et d'un inverseur.

La table de vérité d'un demi-additionneur capable d'effectuer l'addition de deux chiffres binaires est donnée à la figure 2.

Les équations booléennes décrivant le fonctionnement d'un demi-additionneur sont les suivantes :

$$S = \bar{A}B + A\bar{B}$$

$$R = AB$$

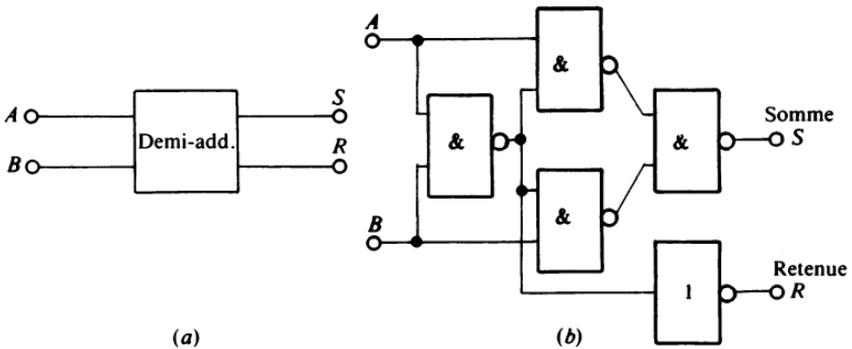


Fig. 1 Demi-additionneur :

- (a) symbole logique,
 (b) logigramme.

A	B	S	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Fig. 2 Table de vérité d'un demi-additionneur.

Le demi-additionneur ne peut faire l'addition que du chiffre de plus faible poids de deux nombres, puisqu'il ne peut pas prendre en compte la retenue qui proviendrait d'une colonne précédente. L'additionneur complet supprime cette limitation car il possède trois entrées, c'est-à-dire qu'en plus des deux chiffres à additionner, il prend en compte la retenue provenant d'une colonne précédente. La figure 3 montre le principe d'un additionneur complet, constitué de deux demi-additionneurs.

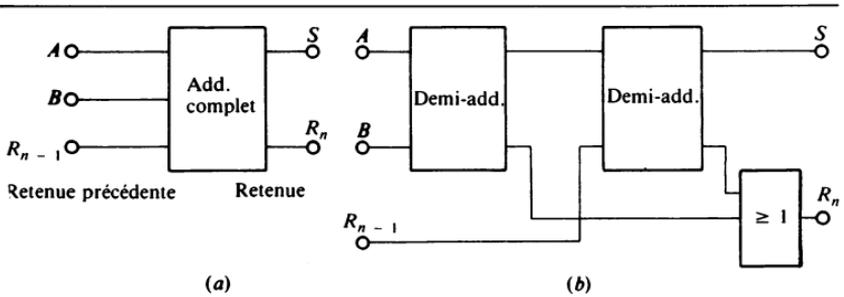


Fig. 3 Additionneur complet :
 (a) symbole logique,
 (b) utilisant deux demi-additionneurs.

La table de vérité d'un additionneur complet capable d'effectuer l'addition de deux chiffres binaires et d'une retenue est donnée à la figure 4.

A	B	R_{n-1}	S	R_n
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Fig. 4 Table de vérité d'un additionneur complet.

nant en compte la retenue provenant de la colonne précédente. Il faut envisager maintenant le processus complet permettant d'additionner deux nombres binaires.

Dans l'addition série, les chiffres à additionner sont présentés séquentiellement, c'est-à-dire les uns après les autres, à un additionneur complet. La figure 6 illustre le fonctionnement d'un additionneur série.

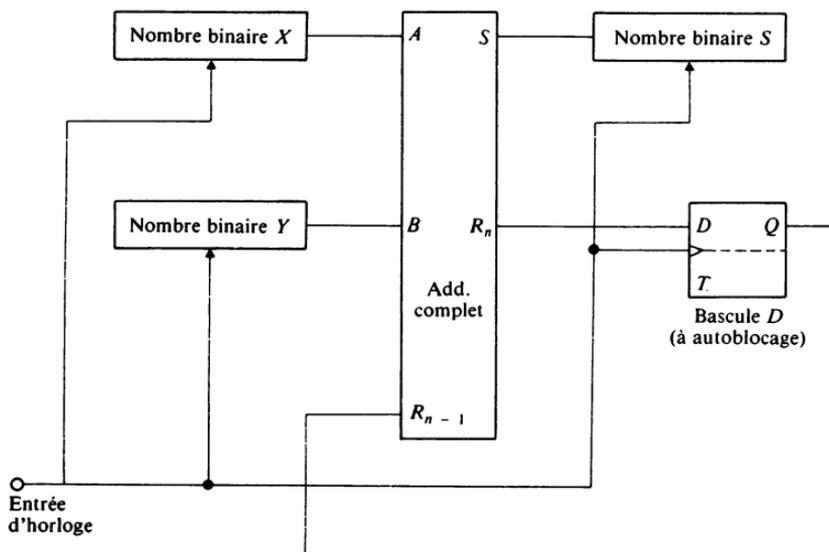


Fig. 6 Additionneur série.

Les nombres binaires X , Y et la somme S sont stockés dans les registres à décalage (voir chapitre 7). Chaque chiffre des nombres X et Y est décalé séquentiellement dans l'additionneur, sous la commande d'impulsions d'horloge (H), en commençant par le chiffre de plus faible poids. En utilisant la même horloge pour commander l'entrée des données dans l'additionneur et dans le registre de la somme, ainsi que le fonctionnement de la bascule D , on s'affranchit des problèmes de temps de propagation de la retenue à travers la bascule D .

En supposant que la sortie Q de la bascule D est initialement à l'état « 0 », l'application de la première impulsion d'horloge a pour effet de présenter le chiffre de plus faible poids des nombres X et Y aux entrées A et B de l'additionneur, la somme S apparaît en sortie; la retenue est appliquée à l'entrée de la bascule D . A l'impulsion d'horloge suivante, le chiffre suivant des nombres X et Y est présenté aux entrées A et B de l'additionneur complet, la somme précédente S est décalée dans le registre à décalage, la retenue précédente est enregistrée par la bascule et se présente alors à l'entrée R_{n-1} de l'additionneur complet. Cette procédure se répète

jusqu'à la fin de l'addition des chiffres des deux nombres X et Y , et le résultat final de la somme se trouve alors dans le registre S .

L'additionneur parallèle

L'additionneur parallèle est capable de fournir quasi instantanément le résultat de l'addition. Toutefois, cette performance n'est possible que grâce à un accroissement du nombre de circuits. L'additionneur parallèle nécessite un additionneur complet pour chaque chiffre à traiter. Un exemple d'additionneur parallèle est donné à la figure 7; il est capable d'additionner deux nombres binaires à quatre chiffres à l'aide de quatre additionneurs complets.

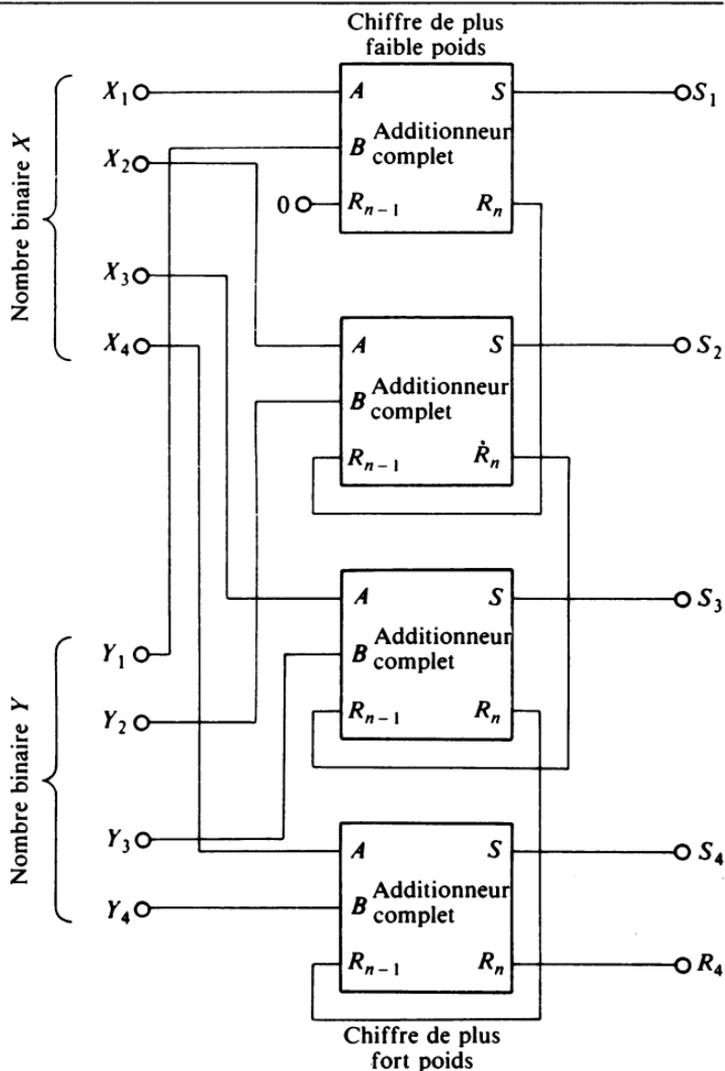


Fig. 7 Additionneur parallèle (à 4 chiffres).

Ce montage est réalisé sous la forme d'un circuit intégré. La figure 8 montre un circuit SN 7483, additionneur complet à quatre bits dans lequel les deux nombres à additionner sont désignés par A et B et la somme par Σ .

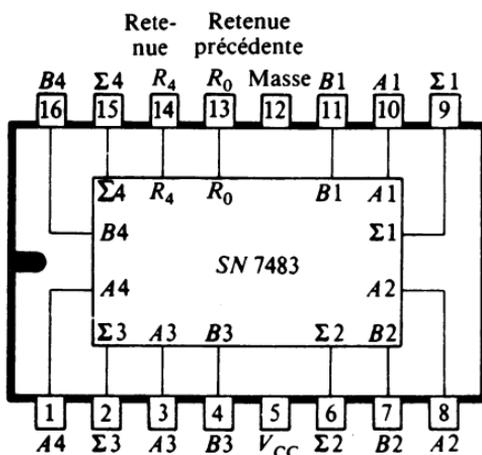


Fig. 8 Additionneur complet à 4 bits (SN 7483).

Le temps total nécessaire à cette addition parallèle, correspond au temps de propagation de la retenue à travers les circuits de l'additionneur.

La soustraction binaire

Différentes méthodes permettent de réaliser la soustraction de deux nombres binaires : la méthode arithmétique et la méthode du complément à deux.

LA MÉTHODE ARITHMÉTIQUE

Cette méthode est la même que celle permettant d'effectuer une soustraction décimale. En commençant par le chiffre de plus faible poids (LSB), situé à l'extrême droite, on soustrait les « 1 » de la façon habituelle :

- si on doit soustraire un 0 d'un 0, le résultat est égal à « 0 », et on ne reporte aucune retenue dans la colonne suivante.
- si on doit soustraire un 0 d'un 1, le résultat est égal à « 1 », et on ne reporte aucune retenue dans la colonne suivante.
- si on doit soustraire un 1 d'un 0, le résultat est égal à « 1 », et on reporte un « 1 » de retenue dans la colonne suivante.
- si on doit soustraire un 1 d'un 1, le résultat est égal à « 0 », et on ne reporte aucune retenue dans la colonne suivante.

EXEMPLE

Soit la soustraction binaire de deux nombres décimaux 40 et 22 :

$$\begin{array}{r} 101000 = 40 \\ - 10110 = 22 \\ \hline 10010 = 18 \end{array}$$

MÉTHODE DU COMPLÉMENT À DEUX

Cette méthode est la plus employée, car son application revient à faire l'addition du complément à deux du nombre que l'on devait soustraire.

Le complément à deux d'un nombre binaire à soustraire se réalise en s'assurant tout d'abord que ce nombre a le même nombre de chiffres que le nombre duquel il doit être soustrait. Ensuite, il est complémenté, c'est-à-dire que tous les « 0 » sont changés en « 1 » et tous les « 1 » en « 0 ». Et finalement, on lui ajoute un « 1 » au chiffre de plus faible poids.

EXEMPLE

Soit la soustraction binaire des deux nombres décimaux 40 et 22 :

$$\begin{array}{l} 40 = 101000 \\ 22 = 10110 \end{array}$$

On s'assure, tout d'abord, que le nombre à soustraire a autant de chiffres que le nombre duquel il est à soustraire. Dans le cas présent, on lui rajoute un chiffre, car il n'en possède que 5 (contre 6 pour le premier nombre).

$$10110 = 010110$$

On complémenté le nombre ainsi trouvé, c'est-à-dire qu'on transforme les « 0 » en « 1 » et vice versa :

$$\begin{array}{l} 010110 \\ 101001 \end{array}$$

On ajoute « 1 » au chiffre de plus faible poids du nombre complémenté :

$$\begin{array}{r} 101001 \\ + \quad 1 \\ \hline 101010 \end{array}$$

Finalement, on additionne ce nombre avec celui duquel il fallait le soustraire initialement :

$$\begin{array}{r} 101000 = 40 \\ + 101010 = \text{complément à deux de 22} \\ \hline 1010010 \quad 18 \end{array}$$

On ne conservera que les six derniers chiffres du résultat, le 1 situé dans la colonne d'extrême gauche est un 1 de débordement, et on ne doit pas en tenir compte.

Le soustracteur binaire

La soustraction binaire d'un nombre Y d'un nombre X pouvant se faire par l'addition du complément à deux de Y , on conçoit aisément

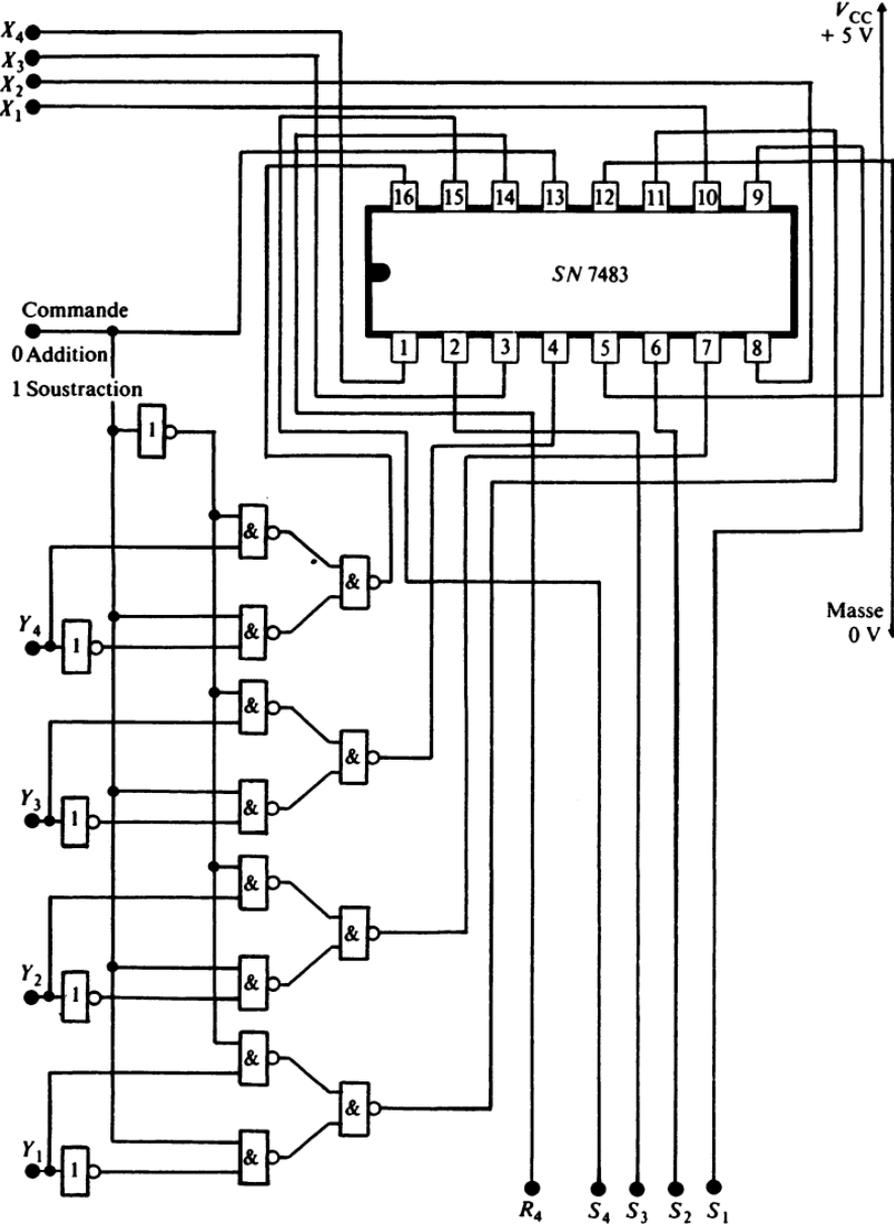


Fig. 9 Additionneur-soustracteur à 4 bits

ment qu'une modification de l'additionneur parallèle doit pouvoir lui permettre de réaliser également des soustractions. Il suffit pour cela d'inverser chaque bit du nombre Y à l'aide d'un inverseur (NON) et de rajouter « 1 » sur l'entrée R_0 de retenue précédente de l'additionneur.

La figure 9 montre un exemple d'additionneur-soustracteur à quatre bits, dont la fonction d'addition ou de soustraction se commande grâce à un niveau logique que l'on applique sur une entrée de commande, un « 0 » commandant l'addition et un « 1 » commandant la soustraction.

MONTAGE PRATIQUE

Additionneur parallèle à 4 éléments binaires (bits)

- Réaliser le montage de l'additionneur parallèle à 4 bits de la figure 7, en utilisant un circuit SN 7483 (additionneur complet à 4 éléments binaires) décrit à la figure 8.
- Déterminer la valeur des nombres binaires X et Y . Après mise sous tension, examiner la valeur de la somme S et de la retenue R_4 . Confirmer ces résultats à l'aide de la méthode arithmétique. Répéter plusieurs fois cet exercice en donnant d'autres valeurs à X et à Y .

MONTAGE PRATIQUE

Additionneur-soustracteur à 4 éléments binaires (bits)

- Réaliser le montage de l'additionneur-soustracteur à 4 éléments binaires de la figure 9, en utilisant 1 circuit SN 7483 (additionneur complet à 4 bits), 3 circuits SN 7400 (quadruple porte EN-NON à 2 entrées) et 5/6 de circuit SN 7404 (sextuple inverseur).
- Déterminer la valeur des nombres binaires X et Y . Après mise sous tension du montage, appliquer un niveau logique « 0 » sur l'entrée de commande, et examiner la valeur des sorties S_1 à S_4 , et s'assurer que l'opération réalisée est bien une addition. De même, appliquer un niveau logique « 1 » sur l'entrée de commande et surveiller la valeur des sorties S_1 à S_4 , pour confirmer que l'opération est bien une soustraction. Répéter plusieurs fois cet exercice en donnant d'autres valeurs à X et à Y .

La multiplication binaire

La multiplication binaire se réalise simplement en effectuant une addition de nombres binaires décalés, puisqu'il n'est nécessaire que de multiplier par « 0 » ou par « 1 ». L'exemple suivant explique le mécanisme d'une multiplication binaire.

EXEMPLE

Soit la multiplication binaire des deux nombres décimaux 11 et 5.

$$\begin{array}{r} 1011 = 11 \\ \times 101 = 5 \\ \hline 1011 \\ 0000 \\ 1011 \\ \hline 110111 = 55 \end{array}$$

Le logigramme d'un circuit capable de multiplier un membre binaire X à 4 éléments binaires (bits) par un nombre binaire Y à 3 bits est donné à la figure 10.

Les chiffres X_1 à X_4 sont commandés par les niveaux logiques des chiffres Y_1 à Y_3 , et chaque fois que l'on passe de Y_1 à Y_2 ou de Y_2 à Y_3 , les chiffres X_{11} à X_{41} sont décalés d'un chiffre binaire, lorsqu'ils arrivent aux additionneurs complets.

MONTAGE PRATIQUE

- Réaliser le montage du multiplicateur parallèle de la figure 10, en utilisant 2 circuits SN 7483 (additionneur complet à 4 bits), 3 circuits SN 7400 (quadruple porte ET-NON à 2 entrées) et 2 circuits SN 7404 (sextuple inverseur).
- Déterminer la valeur des 4 bits du nombre X et des 3 bits du nombre Y . Après mise sous tension, examiner la valeur des sorties S_1 à S_6 . Vérifier le produit de X par Y par la méthode arithmétique. Répéter plusieurs fois cet exercice en donnant d'autres valeurs à X et à Y .

La division binaire

Comme la multiplication binaire, la division se réalise en effectuant une suite de soustractions et de décalages. L'exemple suivant en explique le mécanisme.

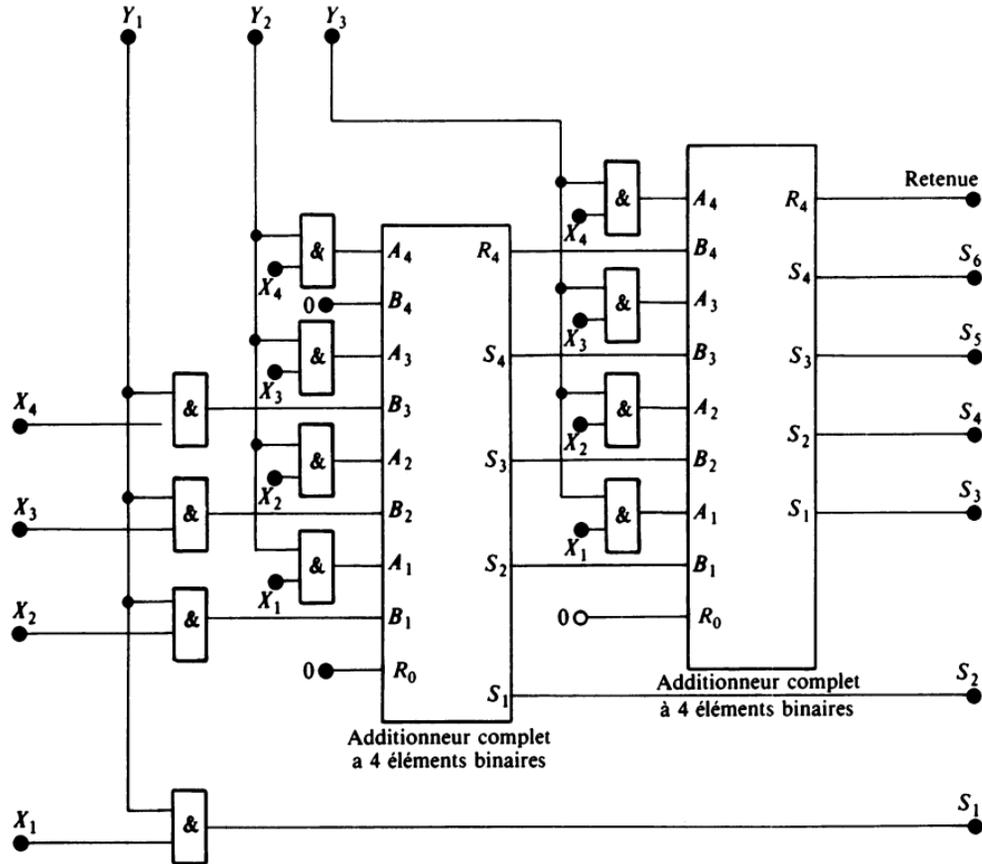
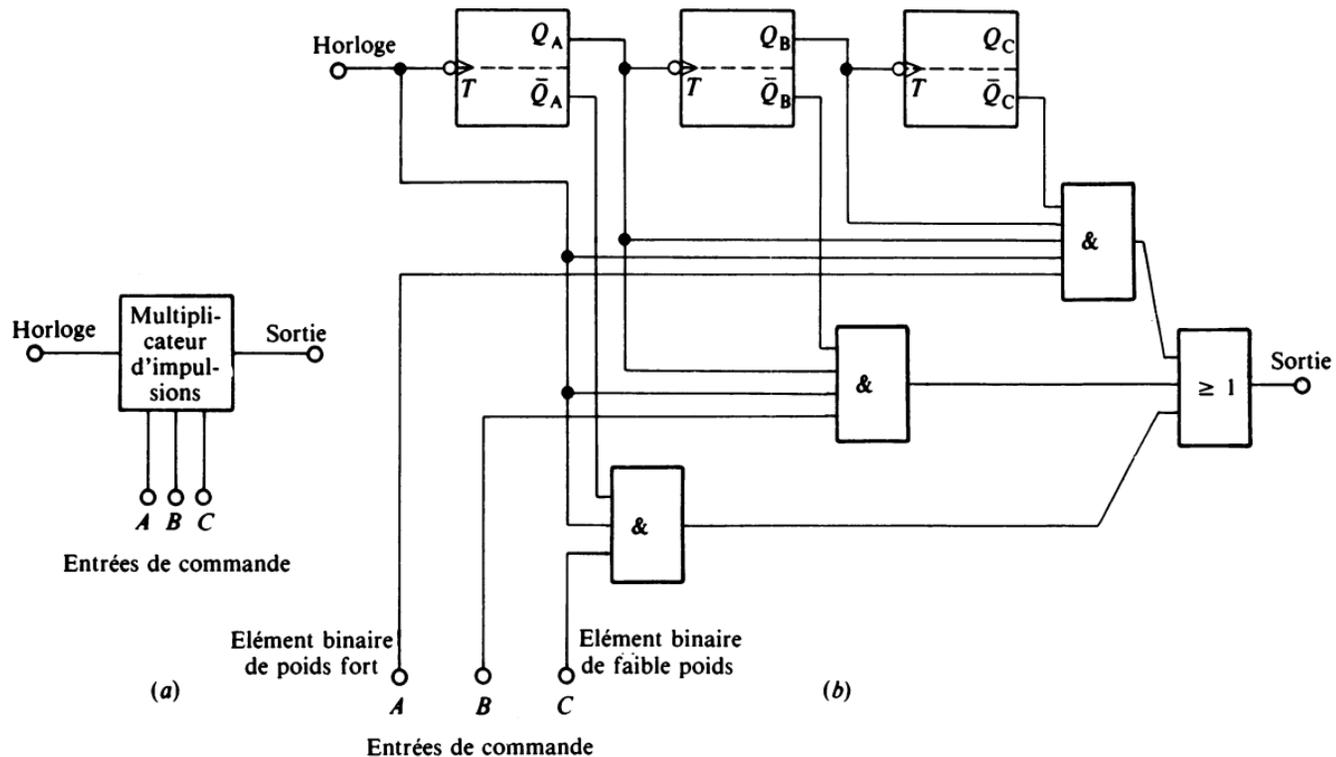


Fig. 10 Multipliqueur parallèle à 4×3 bits.



Entrées de commande

Fig. 11 Multiplicateur d'impulsions à 3 éléments binaires :

(a) symbole logique,
(b) logigramme.

EXEMPLE

Soit à diviser les deux nombres décimaux 55 et 5.

$$\begin{array}{r|l} 110111 (55) & 101 (5) \\ - 101 & \\ \hline 0011 & 1011 (11) \text{Quotient} \\ - 000 & \\ \hline 0111 & \\ - 101 & \\ \hline 0101 & \\ - 101 & \\ \hline 000 & \text{Reste} \end{array}$$

Les logigrammes permettant d'effectuer une division binaire s'apparentent aux logigrammes précédents de la multiplication et de la soustraction.

Le multiplicateur d'impulsions

Le multiplicateur d'impulsions est un circuit capable de fournir en sortie un train d'impulsions dont le nombre varie en fonction des signaux d'entrée A , B et C . La figure 11 montre un multiplicateur d'impulsions à 3 bits.

Après quelques modifications, un multiplicateur d'impulsions peut réaliser certaines opérations mathématiques, tel que l'addition, la soustraction, la multiplication, la division, l'exponentielle, l'intégration, et quelques autres fonctions mathématiques complexes (Unité arithmétique et logique).

des registres sur qui compter

les registres et les compteurs

Lorsque l'on relie ensemble un groupe de bascules pour enregistrer une information, on crée un registre.

Un *registre à décalage* est un registre dans lequel les données peuvent être décalées à droite ou à gauche.

Certains types de registres peuvent servir pour compter des impulsions. Ils portent alors le nom de compteurs.

Un compteur Johnson est un registre à décalage bouclé sur lui-même (compteur en anneau).

Le registre-mémoire

Un registre-mémoire peut être constitué par l'assemblage de bascules ayant une entrée d'horloge, tel que les bascules *D* et *JK* décrites au chapitre 6. Un registre-mémoire accepte des données, les mémorise et les restitue lorsque cela est nécessaire.

La figure 1 représente un registre-mémoire à 3 bits, utilisant des bascules *D*, dans lequel les informations entrent et sortent en parallèle. Ce registre-mémoire est un registre à entrée et sortie en parallèle.

Dans ce registre, les données à mémoriser se présentent simultanément aux entrées *D* des bascules, sous la forme d'un nombre binaire. Dès l'application d'une impulsion d'horloge, les données sont enregistrées dans les bascules et sont disponibles aux sorties

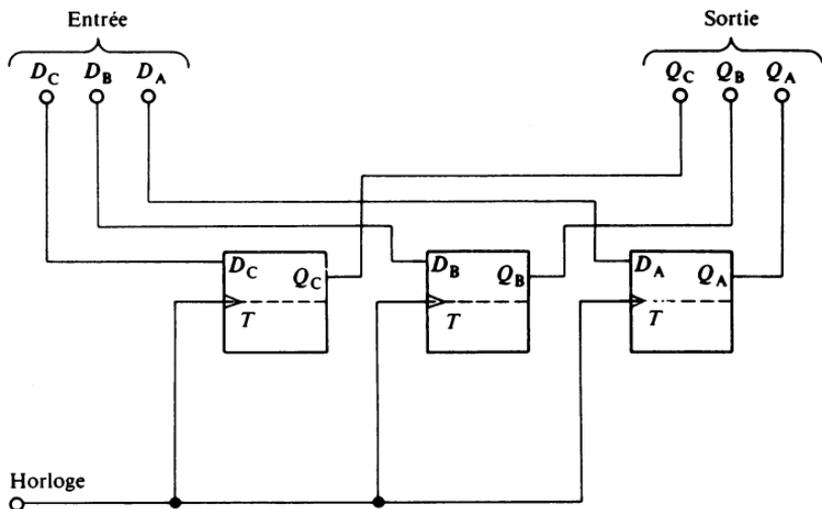


Fig. 1 Registre-mémoire à 3 bits

Q de ces bascules. Il est également possible d'utiliser les entrées de forçage des bascules pour y enregistrer des données, avec, par exemple, le SN 7474 (double bascule D).

REMARQUES

- Avec le circuit SN 7474, lorsque les entrées sont en l'air, c'est-à-dire qu'aucun signal ne leur est appliqué, cela correspond à l'application d'un « 1 » logique.
- Les entrées de forçage sont actives avec un « 0 » logique, et ces entrées ont priorité sur les entrées D et l'entrée d'horloge.

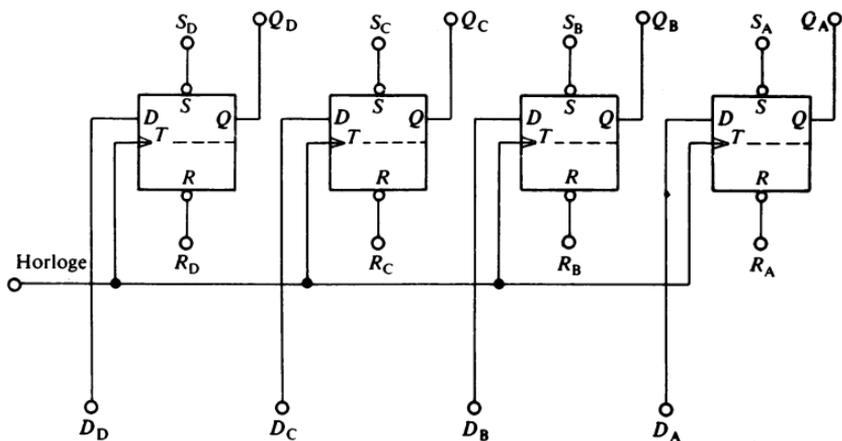


Fig. 2 Logigramme d'un registre-mémoire à 4 bits

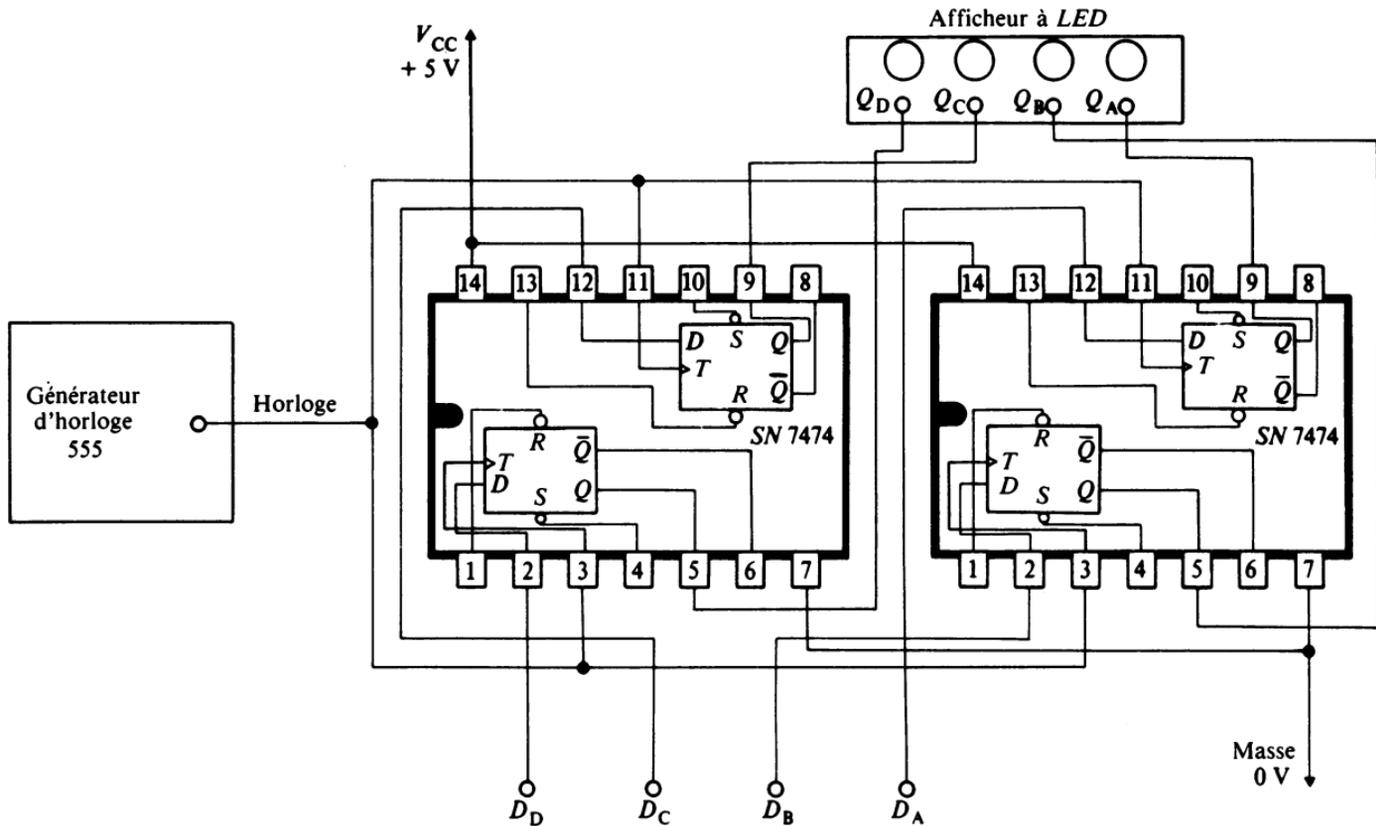


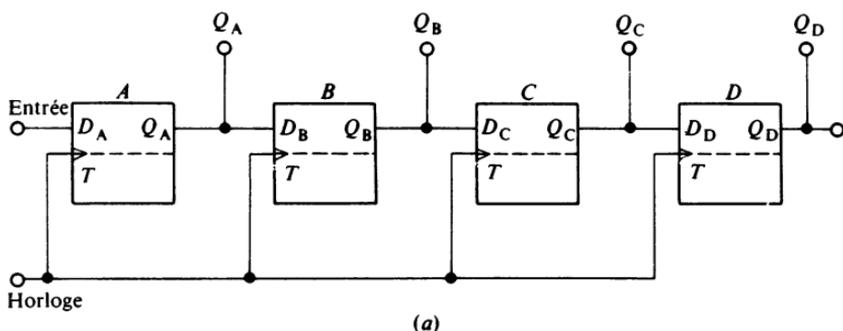
Fig. 3 Schéma de câblage d'un registre-mémoire à 4 bits.

Registre-mémoire à 4 éléments binaires (bits)

- Réaliser le montage du registre-mémoire à 4 bits des figures 2 et 3, en utilisant 2 circuits SN 7474 (double bascule D).
- Appliquer un nombre binaire à 4 bits aux entrées D , le niveau logique 0 étant mis à l'entrée d'horloge. Mettre alors un « 1 » logique à l'entrée d'horloge, de façon à enregistrer le nombre dans le registre, puis appliquer à nouveau un « 0 » logique à cette entrée.
- Observer le contenu du registre grâce aux LED branchées sur les sorties Q des bascules. Répéter plusieurs fois cet exercice en changeant, à chaque fois, la valeur du nombre binaire.

Le registre à décalage

Un registre-mémoire est transformable en un registre à décalage en alimentant l'entrée d'une bascule par la sortie de la précédente, et ainsi de suite. La figure 4 a montre un registre à décalage à bascules D .



Impulsions d'horloge	Q_A	Q_B	Q_C	Q_D
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	0	0	0	0

(b)

Fig. 4 Registre à décalage à droite :

- (a) logigramme,
(b) table de vérité.

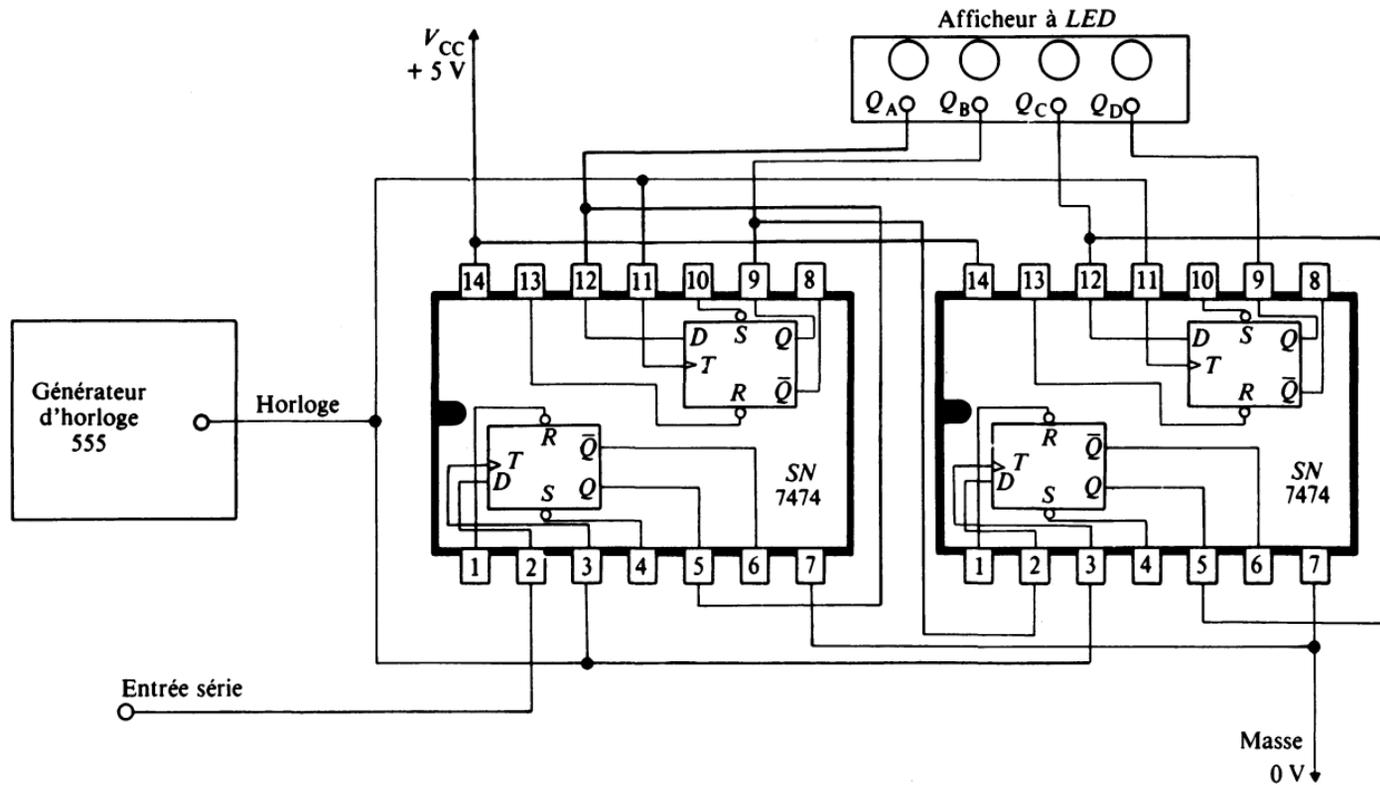


Fig. 5 Schéma de câblage d'un registre à décalage à droite.

Si l'on désire décaler un « 1 » logique dans ce registre, bascule après bascule, on doit d'abord appliquer un « 1 » logique sur l'entrée D_A . Dès la première impulsion d'horloge, on remplace le « 1 » par un « 0 » logique, qu'on maintient jusqu'à la fin de la séquence. A chaque impulsion d'horloge, le « 1 » logique se propage d'une bascule vers la droite, tel que l'indique la figure 4(b).

MONTAGE PRATIQUE

Registre à décalage à droite

- Réaliser le montage des figures 4a et 5, en utilisant 2 circuits SN 7474 (double bascule D).
- Brancher le générateur d'horloge 555 sur l'entrée d'horloge et le régler sur une fréquence très lente. Appliquer un « 0 » logique sur l'entrée et attendre 4 impulsions d'horloge, de façon à vider le registre. Il ne comporte alors que des « 0 ».
- Appliquer un « 1 » logique sur l'entrée, et le maintenir jusqu'à ce que la sortie Q_A indique un niveau « 1 ». Appliquer alors à nouveau, en permanence, un « 0 » logique sur l'entrée. Observer que le « 1 » logique se décale vers la droite à chaque impulsion d'horloge. Au bout de 4 impulsions d'horloge, le registre s'est vidé.

Commande d'écriture d'un registre à décalage

Le registre-mémoire et le registre à décalage peuvent enregistrer des données en parallèle (les éléments binaires sont écrits simultanément), ou en série (les éléments binaires sont écrits séquentiellement à l'entrée de la première bascule), et peuvent restituer les données en parallèle (les éléments binaires sont lus simultanément), ou en série (les éléments binaires sont lus séquentiellement à la sortie de la dernière bascule).

Si le registre utilise des bascules n'ayant pas d'entrée de forçage, la commande d'écriture parallèle des données ne peut se faire que par adjonction de circuits logiques, tel que le montre la figure 6. Lorsque la commande d'écriture est au niveau « 1 », les données parallèles ont accès aux entrées D des bascules, qui les enregistrent dès l'application d'une impulsion d'horloge.

Lorsque la commande d'écriture est au niveau « 0 », les entrées D des bascules sont reliées aux sorties Q des bascules précédentes. Un registre à décalage à entrées parallèles, utilisant des bascules qui possèdent des entrées de forçage est indiqué à la figure 7.

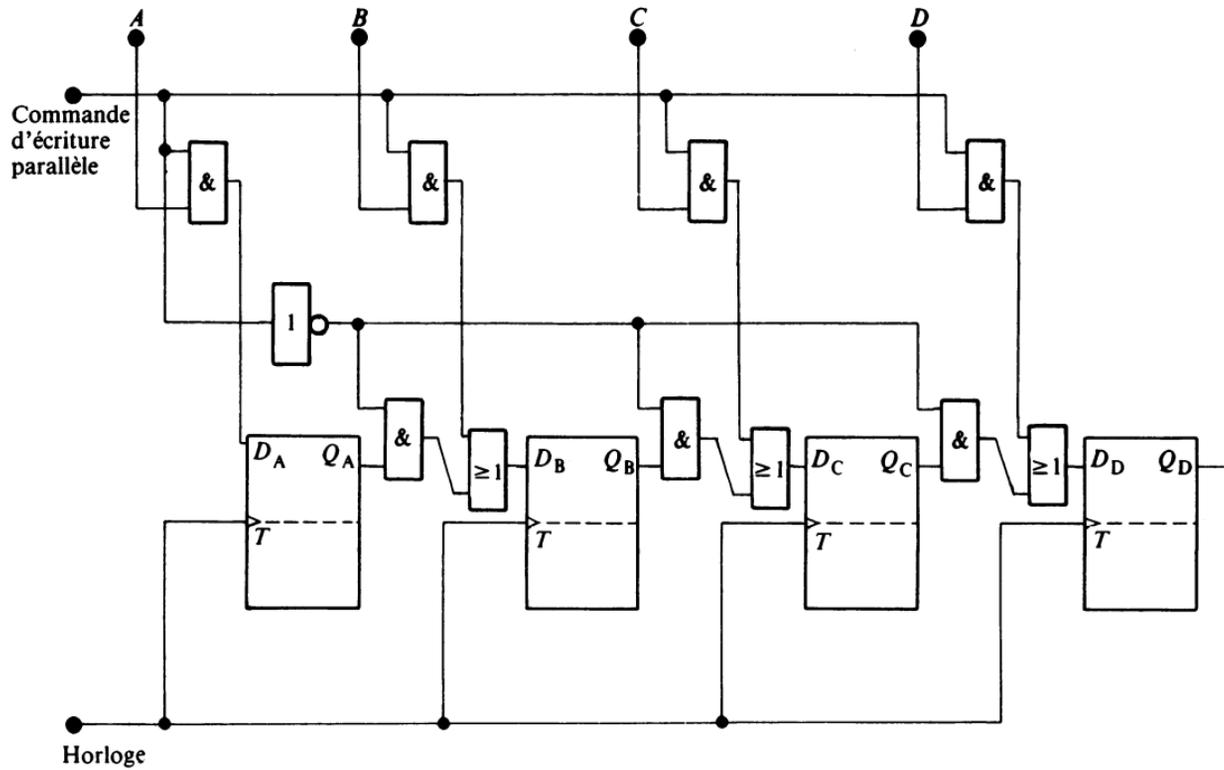


Fig. 6 Commande d'écriture parallèle des données.

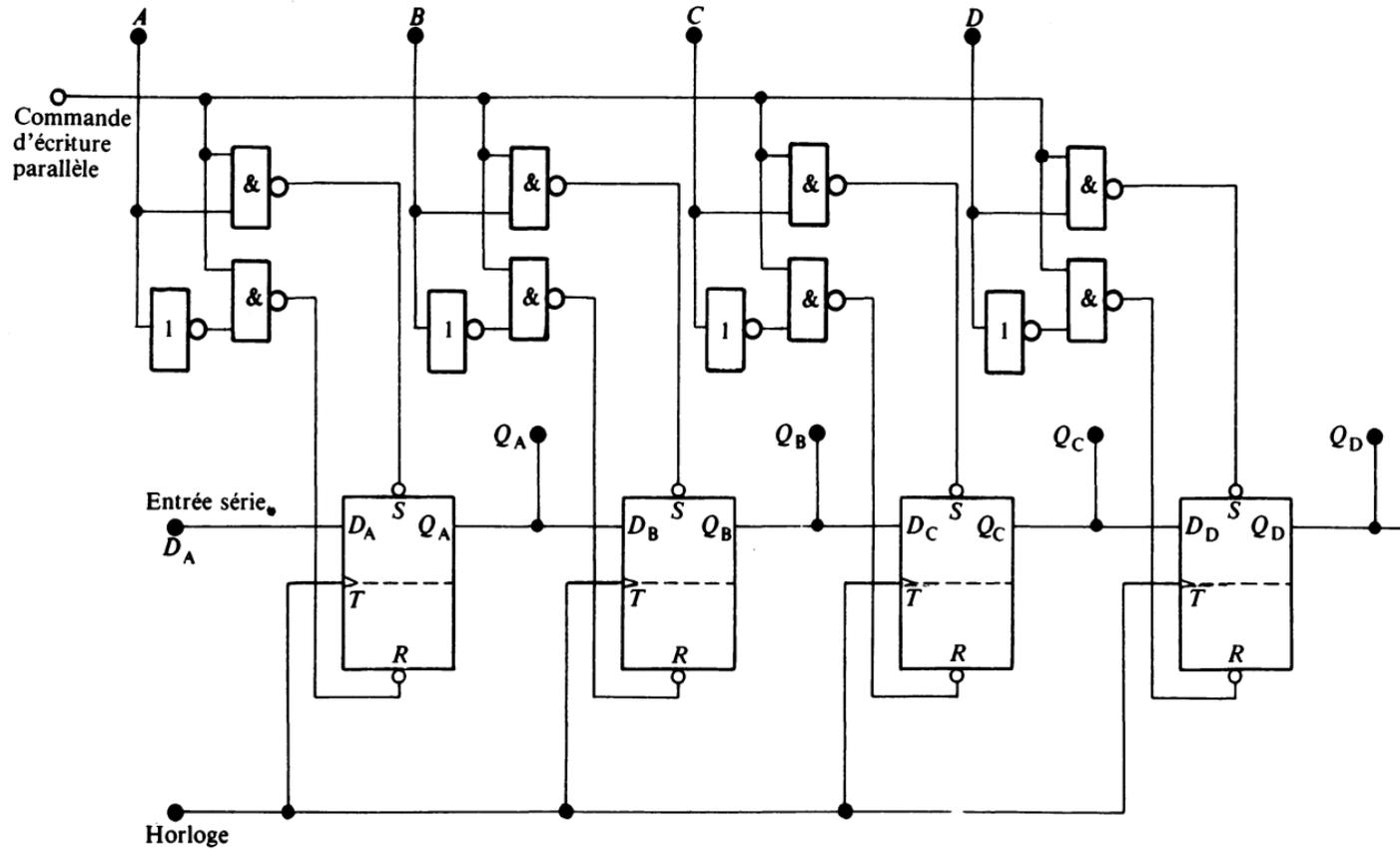


Fig. 7 Registre à décalage à commande d'écriture parallèle.

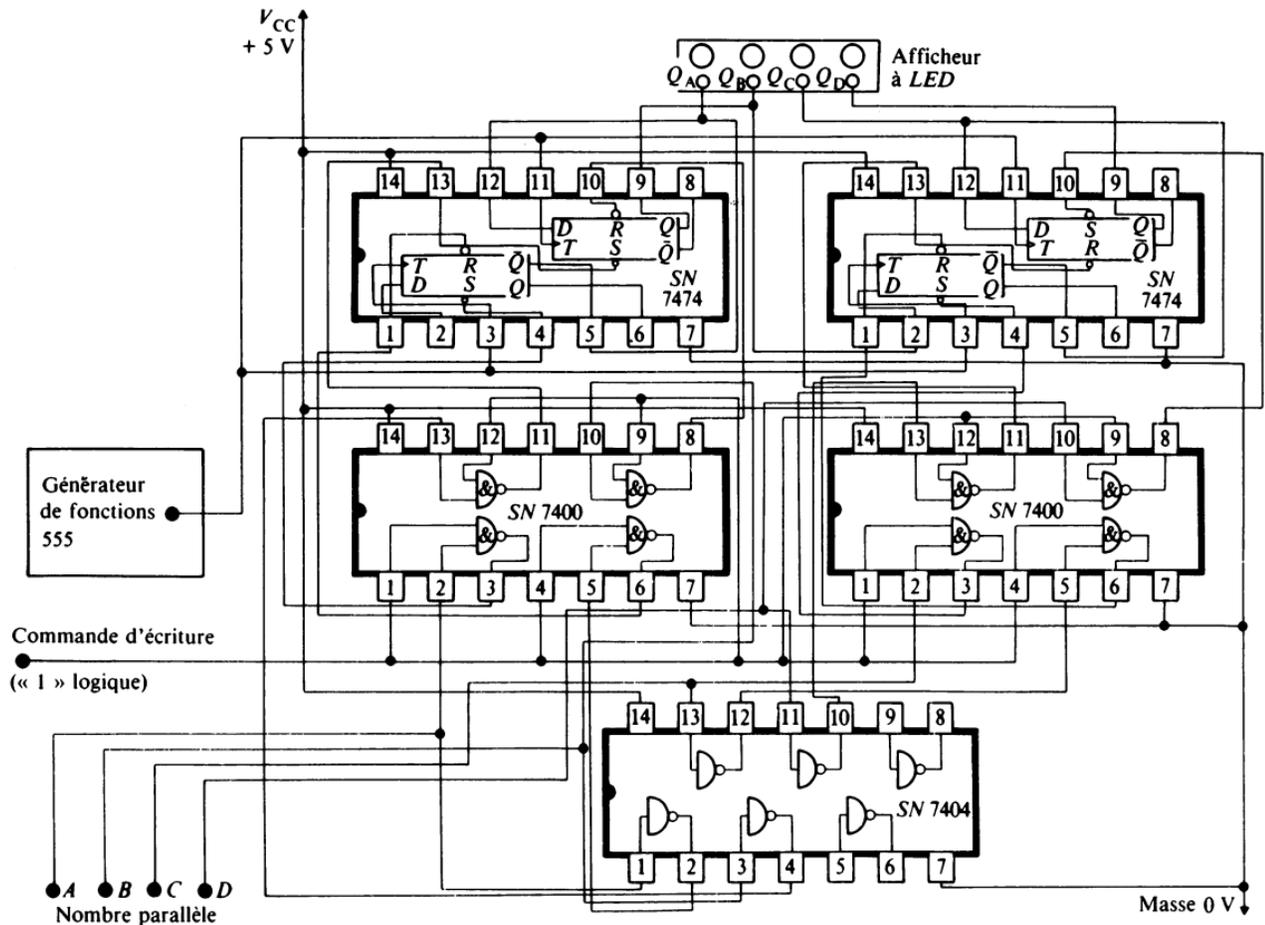


Fig. 8 Schéma de câblage d'un registre à décalage à 4 bits à écriture parallèle.

Dans les deux registres à écriture parallèle décrits aux figures 6 et 7, la lecture parallèle des données se réalise en considérant simultanément les sorties Q des bascules. De plus, une fois les données écrites dans ces registres, elles sont décalées vers la droite à chaque impulsion d'horloge. Le registre indiqué à la figure 7 peut être utilisé en registre à décalage à entrée et sortie en série, en écrivant les données sur l'entrée D_A et en les lisant sur la sortie Q_D .

MONTAGE PRATIQUE

Registre à décalage à entrée et sortie en parallèle

- Réaliser le montage du registre à décalage à entrée et sortie en parallèle des figures 7 et 8, en utilisant 2 circuits SN 7474 (double bascule D), 2 circuits SN 7400 (quadruple porte ET-NON à 2 entrées) et 2/3 de circuits SN 7404 (sextuple inverseur).
- Appliquer un nombre binaire parallèle simultanément sur les entrées A , B , C , et D . Appliquer un « 1 » logique sur la commande d'écriture, et observer la valeur du nombre binaire sur les LED branchées sur les sorties Q_A , Q_B , Q_C et Q_D .
- Appliquer alors un « 0 » logique permanent, et constater le décalage à droite du nombre binaire, au rythme des impulsions d'horloge.

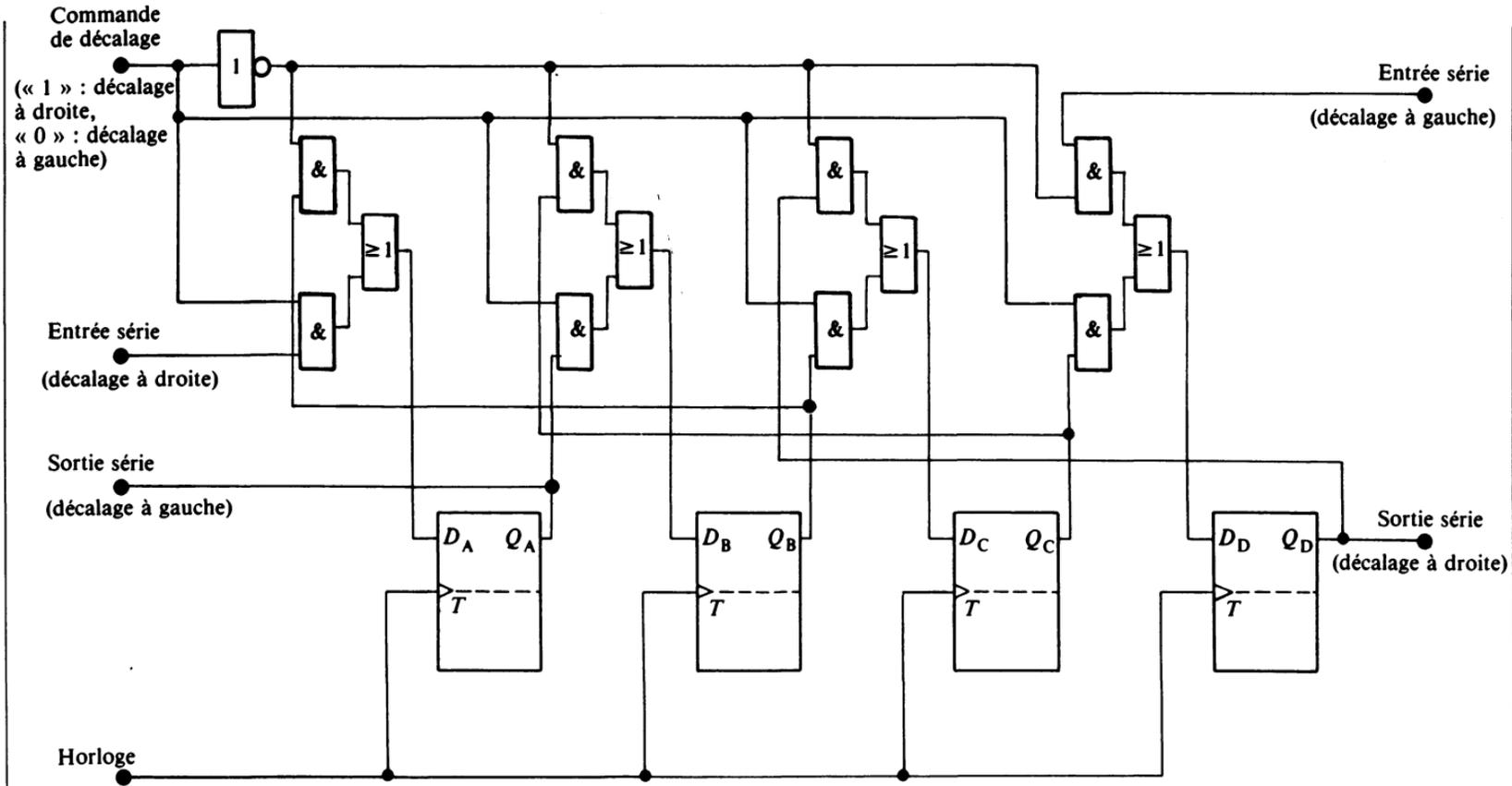
Registre à décalage réversible

Les registres que l'on vient de voir ne sont capables de décaler les données que vers la droite : certaines applications particulières nécessitent un décalage des données soit vers la droite soit vers la gauche. La figure 9 montre comment on peut réaliser un registre à décalage à droite ou à gauche.

La figure 10 montre un exemple de réalisation d'un registre à décalage réversible à entrée et sortie en parallèle et à entrée et sortie en série, utilisant 2 circuits SN7474 de double bascule D , 2 circuits SN 7400 (quadruple porte ET-NON à entrées), 2 circuits SN 7450 (double porte ET-OU-NON à 2 entrées) et 1 1/6 de circuit SN 7404 (sextuple inverseur).

Dans le registre de la figure 10, le nombre binaire à 4 bits est appliqué à l'entrée parallèle. La commande d'écriture est mise à « 1 », ce qui a pour effet d'enregistrer le nombre binaire dans le registre. Il suffit d'appliquer maintenant le niveau logique convenable à la commande de décalage, pour voir les données se décaler à droite ou à gauche au rythme des impulsions d'horloge.

Le nombre binaire à 4 bits peut être appliqué en série sur les entrées séries à décalage à droite ou à gauche; on constate que les



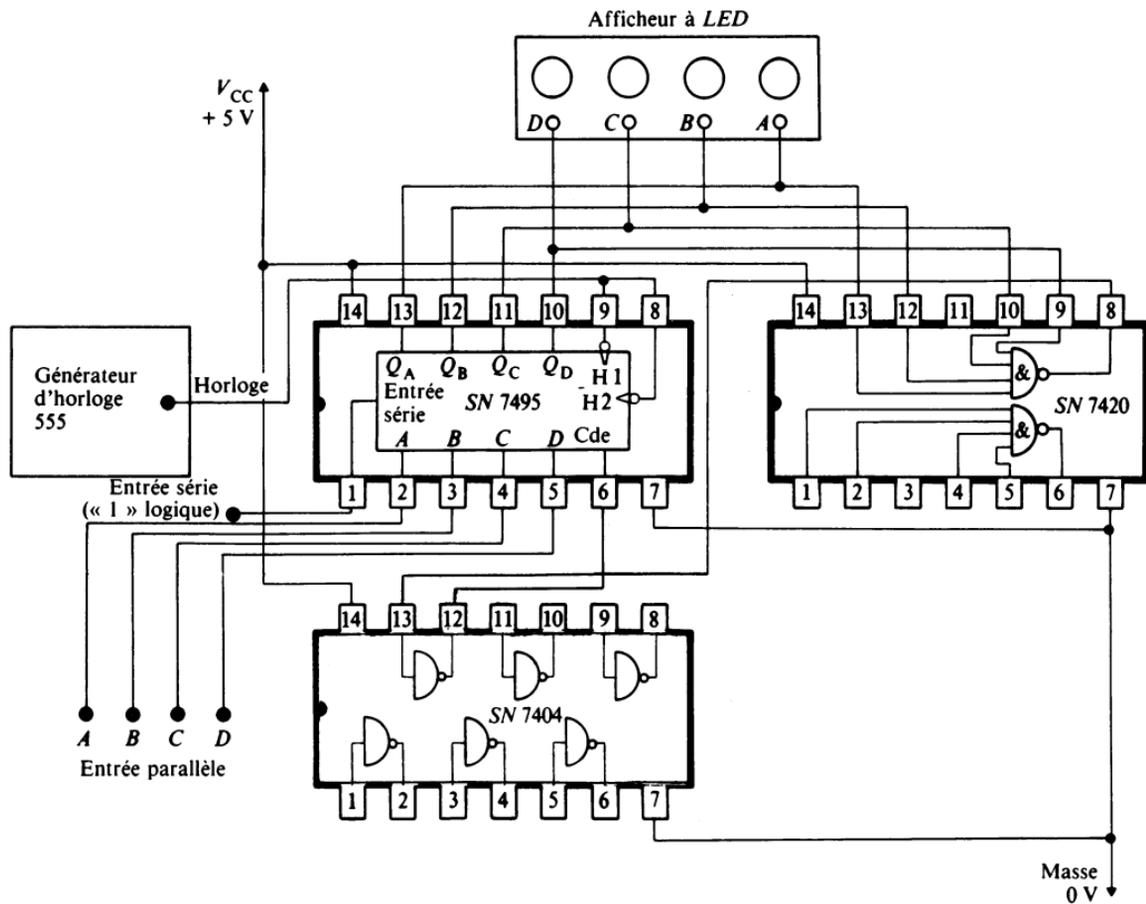


Fig. 11 Convertisseur parallèle-série à 4 bits.

données introduites se décalent au rythme des impulsions d'horloge.

MONTAGE PRATIQUE

Convertisseur parallèle-série à 4 bits

- Réaliser le montage du convertisseur parallèle-série à 4 éléments binaires de la figure 11, en utilisant 1 circuit SN 7495 (registre à décalage à 4 bits, 1/2 circuit SN 7420 (double porte ET-NON à 4 entrées) et 1/6 de circuit SN 7404 (sextuple inverseur).
- Appliquer le nombre binaire à 4 bits à l'entrée parallèle et mettre à « 1 » l'entrée série.
- Brancher le générateur d'horloge 555 sur l'entrée d'horloge et constater que périodiquement le nombre binaire est chargé dans le registre, puis y subit un décalage à droite.

Le compteur asynchrone

Un compteur asynchrone, ou *compteur série*, est un système logique composé de bascules, dans lequel les impulsions que l'on applique à l'entrée, doivent d'abord traverser la première bascule avant de pouvoir commander la seconde, et ainsi de suite jusqu'à la dernière bascule. La méthode la plus simple pour réaliser ce type de compteur consiste à utiliser des bascules JK ou D montées en diviseur par deux, l'entrée d'horloge d'une bascule étant alimentée par la sortie de la bascule précédente. La figure 12 donne un exemple de compteur asynchrone utilisant des bascules JK.

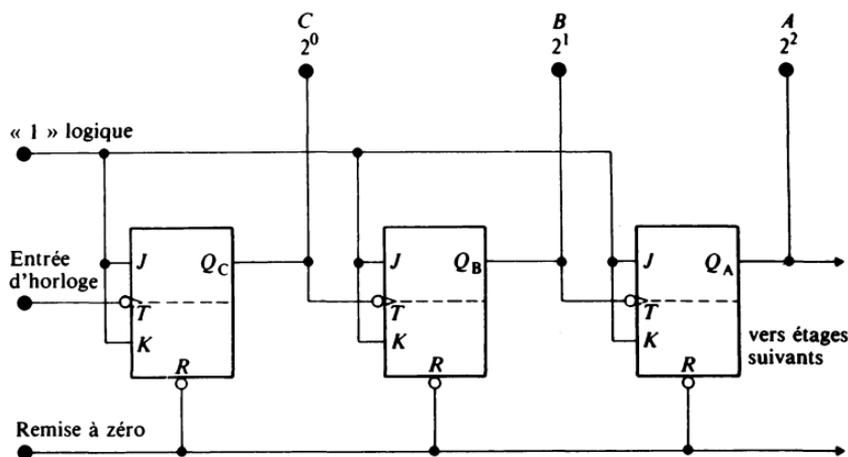


Fig. 12 Compteur asynchrone.

Un tel compteur change d'état au front descendant des signaux appliqués sur les entrées d'horloge. La figure 13 représente le diagramme des temps de ce compteur.

Un tel compteur à trois bascules peut compter de 0 à 7 (en décimal). Dans ce cas, on dit avoir affaire à un compteur *modulo 8*. Bien entendu, on peut compter un plus grand nombre d'impulsions en ajoutant des bascules.

Le compteur-décompteur

On transforme facilement un compteur en décompteur, en inversant les signaux appliqués sur les entrées d'horloge, tel que le montre la figure 14.

MONTAGE PRATIQUE

Le compteur asynchrone

- Réaliser le montage du compteur asynchrone de la figure 15, en utilisant 2 circuits SN 7474 (double bascule *D*).
- Régler la fréquence du générateur d'horloge 555 à une valeur permettant d'observer l'affichage des niveaux binaires par les LED.
- Brancher alors l'entrée d'horloge des bascules sur la sortie *Q* de la bascule précédente, au lieu de la sortie \overline{Q} . On doit observer, grâce aux LED, que l'affichage correspond à celui d'un décompteur, c'est-à-dire un compteur comptant à l'envers.

Le décodage

Il existe diverses applications nécessitant des compteurs dont le modulo n'est pas une puissance de 2, c'est-à-dire 4, 8, 16, 32, etc. Ceci peut être réalisé facilement par un décodeur logique, composé d'une simple porte ET-NON (NAND) détectant l'étape finale de comptage souhaitée. La sortie de cette porte alimente les entrées de remise à zéro des bascules, de façon à remettre le compteur à zéro une fois qu'il a compté le nombre d'impulsions d'horloge souhaitées.

Un exemple de décodage est donné à la figure 16, dans lequel une porte ET-NON à 2 entrées détecte un niveau logique « 1 » sur les sorties Q_B et Q_C , et remet à zéro les bascules *B* et *C* lorsque cela se produit.

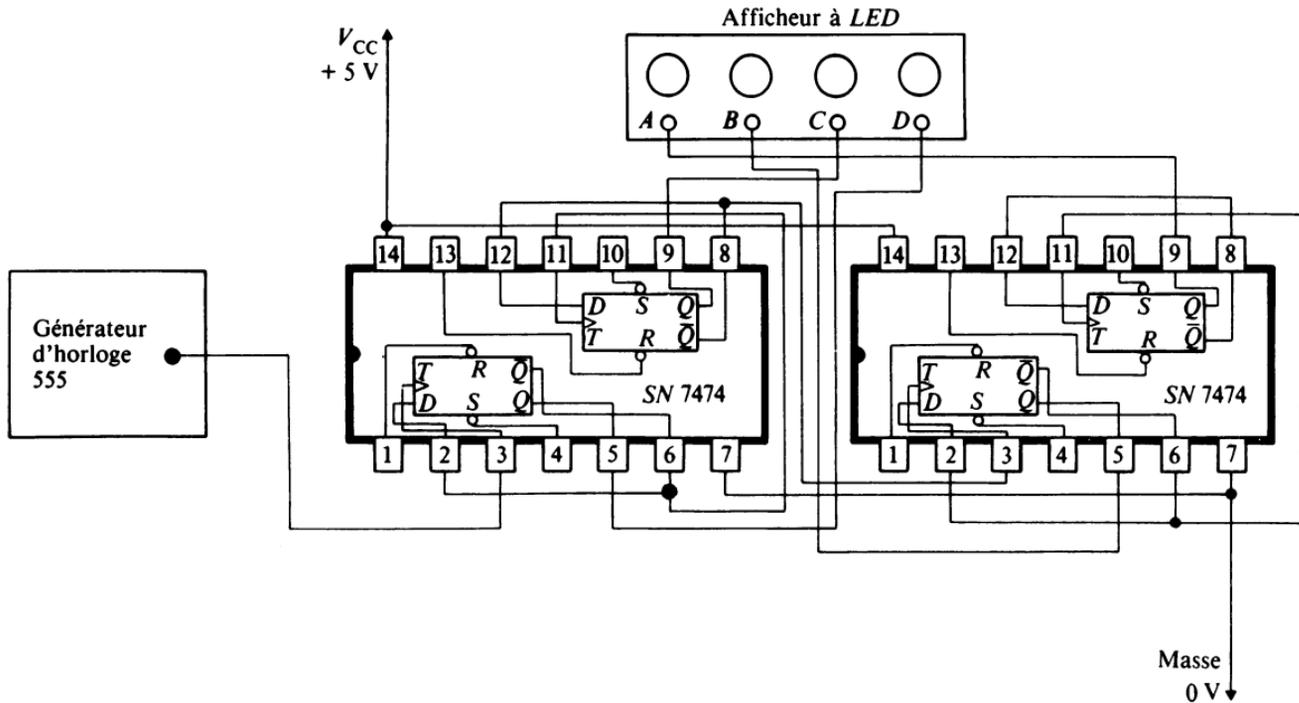


Fig. 15 Compteur asynchrone à 4 bits.

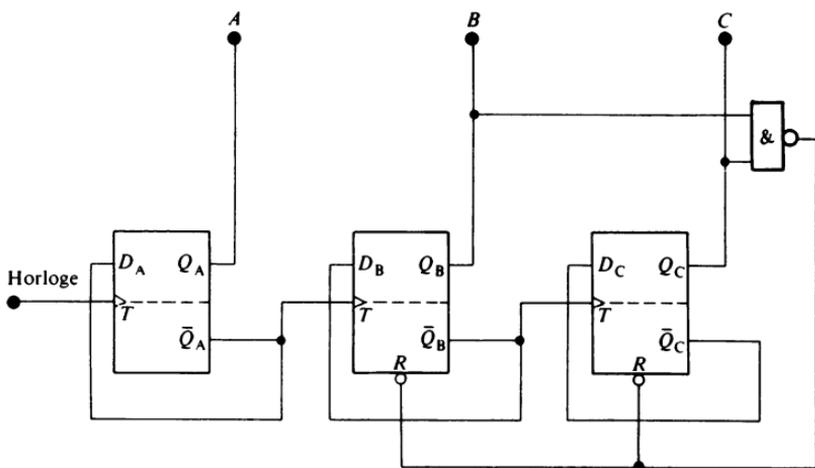


Fig. 16 Décodage.

Une telle disposition modifie le modulo du compteur, puisqu'il est maintenant égal à 6. Cela peut se lire directement sur la figure 16, car la porte ET-NON (NAND) est branchée sur les bascules de poids 2 (bascule B) et de poids 4 (bascule C).

MONTAGE PRATIQUE

Compteur asynchrone de modulo quelconque

- Réaliser un compteur asynchrone de modulo quelconque, en prenant le compteur asynchrone de l'exercice pratique précédent, et en y ajoutant un circuit de décodage, constitué par 1/4 de circuit SN 7400 (quadruple porte ET-NON à 2 entrées), tel que l'indique la figure 17.
- Régler la fréquence du générateur d'horloge 555 à une valeur permettant d'observer l'affichage des niveaux logiques par les LED. Noter le modulo du compteur.
- Trouver la nature et le branchement d'un autre circuit de décodage permettant d'obtenir un modulo différent.

Le compteur synchrone

Un compteur synchrone est un compteur dans lequel une horloge commune est appliquée simultanément à toutes les entrées d'horloge différentes bascules, de façon à ce que ces bascules changent toutes d'état en même temps. Ceci supprime les problèmes dus au

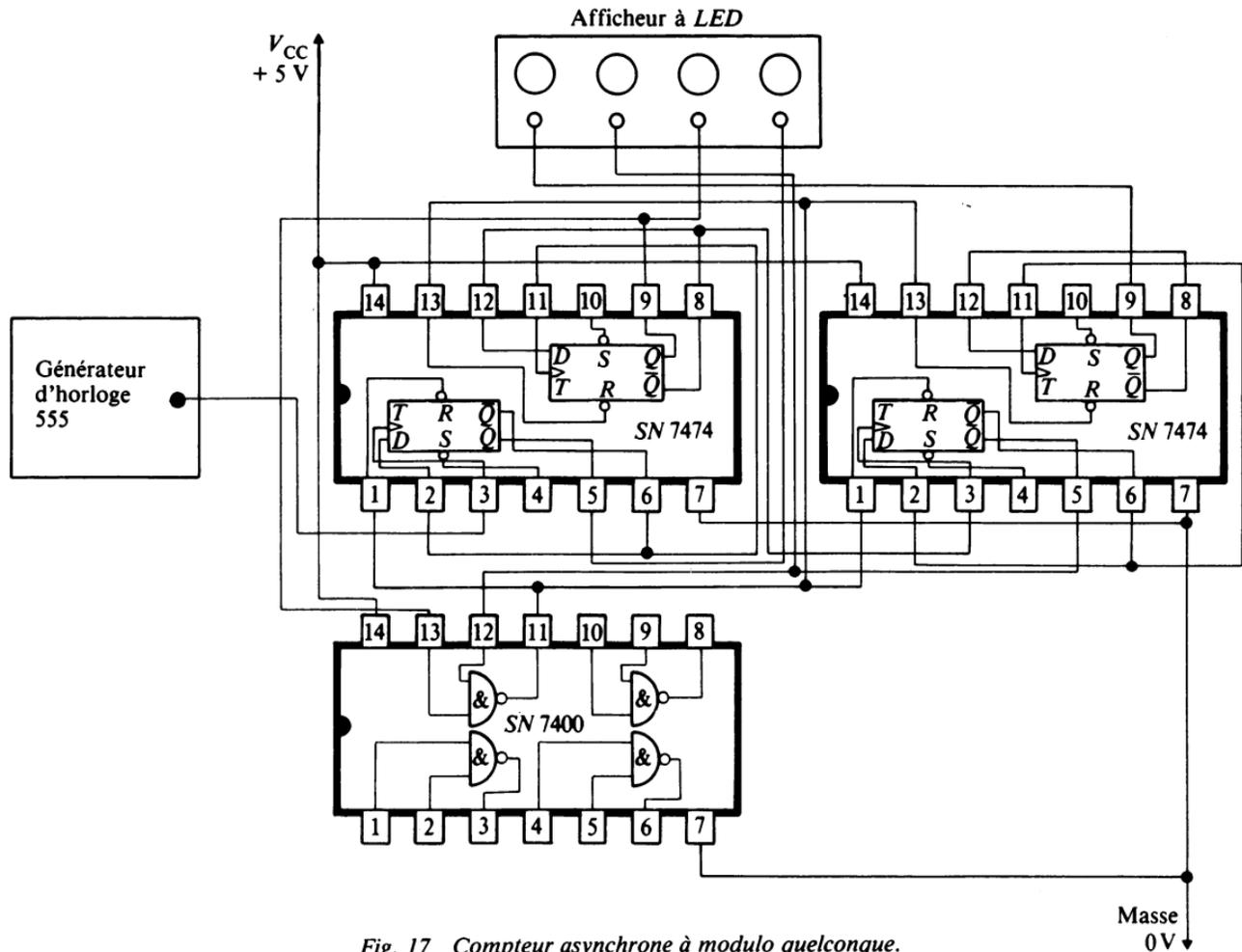


Fig. 17 Compteur asynchrone à modulo quelconque.

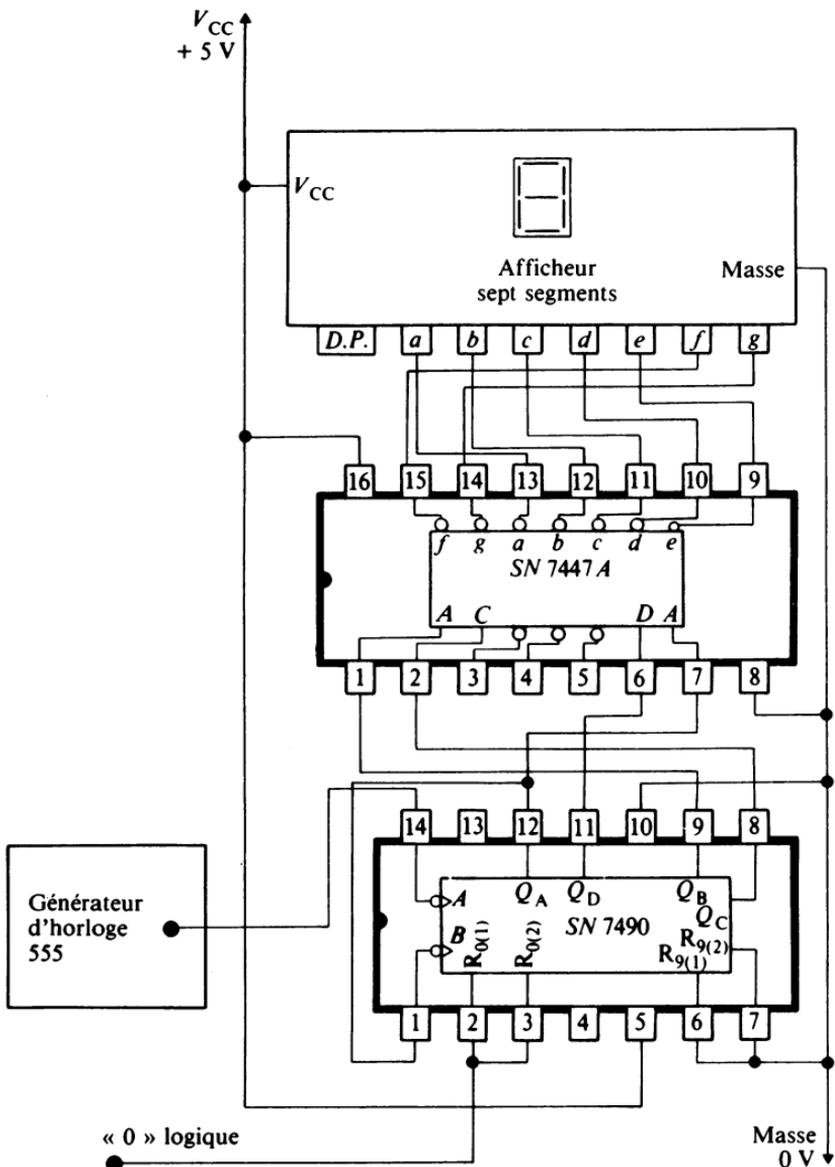


Fig. 19 Décade synchrone.

(décade synchrone), 1 circuit SN 7475 (quadruple bascule D), 1 circuit SN 7447A (décodeur sept segments et 1 afficheur à sept segments.

- Régler la fréquence du générateur d'horloge 555 à une valeur permettant d'observer l'affichage des états du compteur. Appliquer un « 0 » logique sur la commande de mémorisation et noter que l'affichage est maintenu, alors que le compteur continue à compter. Appliquer alors un « 1 » logique sur la commande de mémorisation et noter que l'affichage passe à la valeur atteinte par le compteur pendant le maintien précédent de l'affichage.

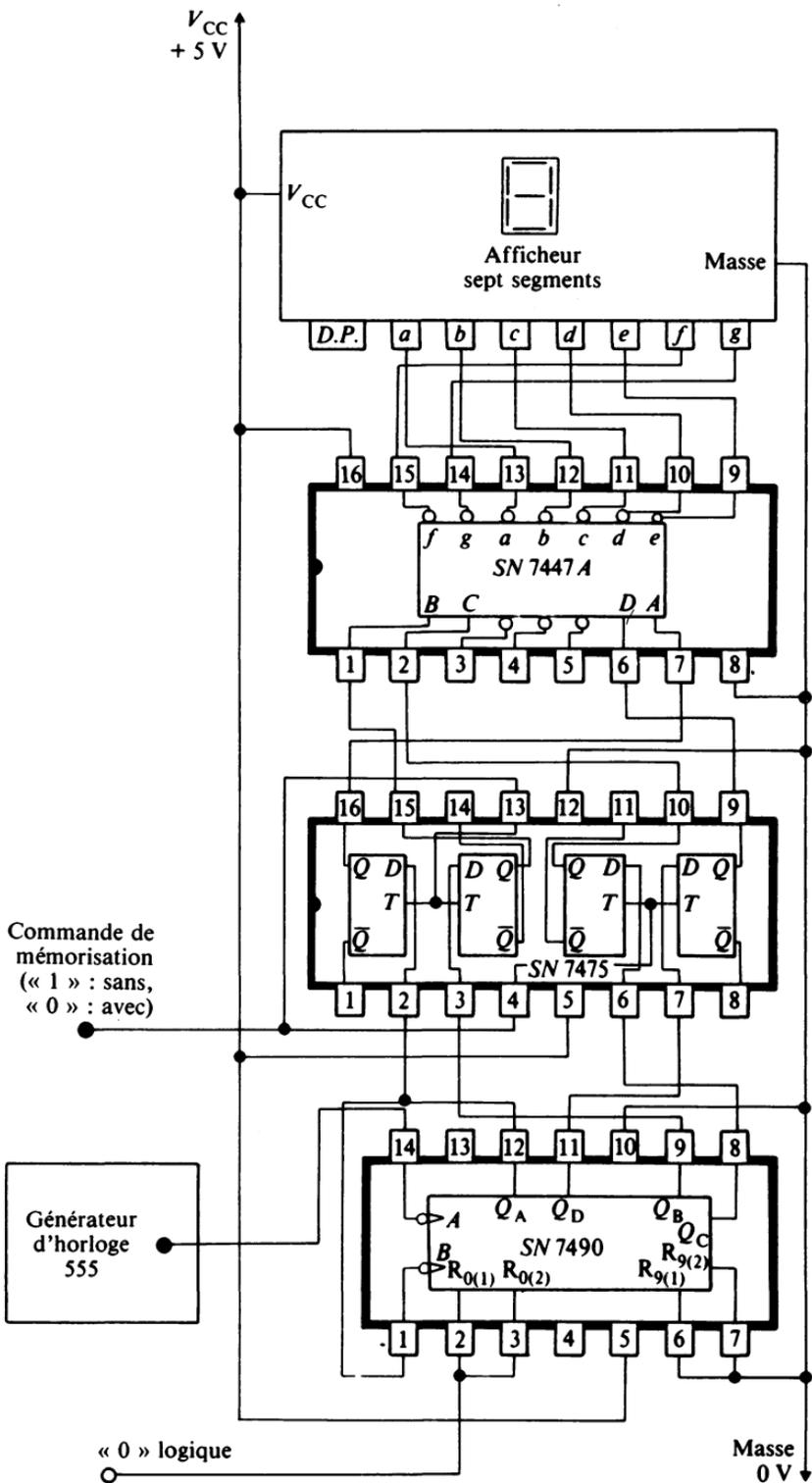


Fig. 20 Décade synchrone avec mémorisation de l'affichage.

Code décimal codé binaire (BCD)

Le système de numération binaire est le système le plus simple et le mieux adapté aux machines. Par contre, le système de numération décimale est le système le plus familier à l'homme. Il convient donc d'établir une méthode simple de conversion du binaire employé par la machine en décimal compréhensible par l'homme, et vice versa.

La méthode binaire conventionnelle (utilisant les puissances de 2) est lourde, car il faut un temps non négligeable pour convertir en décimal une longue suite binaire.

Le système octal simplifie l'écriture du binaire, mais ne permet pas de passer facilement au décimal.

Pour résoudre ce problème, on a créé des codes dans lesquels chaque chiffre décimal correspond à un groupe de 4 éléments binaires (bits), et vice versa, tel que l'indique la figure 21.

Décimal	BCD plus 3	BCD 8421	BCD 2421	BCD 7421
0	0 0 1 1	0 0 0 0	0 0 0 0	0 0 0 0
1	0 1 0 0	0 0 0 1	0 0 0 1	0 0 0 1
2	0 1 0 1	0 0 1 0	0 0 1 0	0 0 1 0
3	0 1 1 0	0 0 1 1	0 0 1 1	0 0 1 1
4	0 1 1 1	0 1 0 0	0 1 0 0	0 1 0 0
5	1 0 0 0	0 1 0 1	0 1 0 1	0 1 0 1
6	1 0 0 1	0 1 1 0	0 1 1 0	0 1 1 0
7	1 0 1 0	0 1 1 1	0 1 1 1	1 0 0 0
8	1 0 1 1	1 0 0 0	1 1 1 0	1 0 0 1
9	1 1 0 0	1 0 0 1	1 1 1 1	1 0 1 0

Fig. 21 Notation BCD (décimal codé binaire).

Les différents codes BCD (décimal codé binaire) sont caractérisés par les différences suivantes :

- **BCD plus 3.** Ce code s'obtient en rajoutant 3 à la valeur décimale à convertir. Les poids des colonnes allant de gauche à droite sont respectivement 8, 4, 2 et 1.
- **BCD 8421.** Les poids des colonnes allant de gauche à droite sont respectivement 8, 4, 2 et 1.
- **BCD 2421.** Les poids des colonnes allant de gauche à droite sont respectivement 2, 4, 2 et 1.
- **BCD 7421.** Les poids des colonnes allant de gauche à droite sont respectivement 7, 4, 2 et 1.

EXEMPLE

Le nombre décimal 79 donne en représentation BCD 8421 :

<u>80</u>	<u>40</u>	<u>20</u>	<u>10</u>	<u>8</u>	<u>4</u>	<u>2</u>	<u>1</u>
0	1	1	1	1	0	0	1

Compteur en BCD 8421

Le logigramme d'un compteur asynchrone en BCD 8421 est donné à la figure 22; les différents états de ces bascules correspondent au tableau de la figure 21.

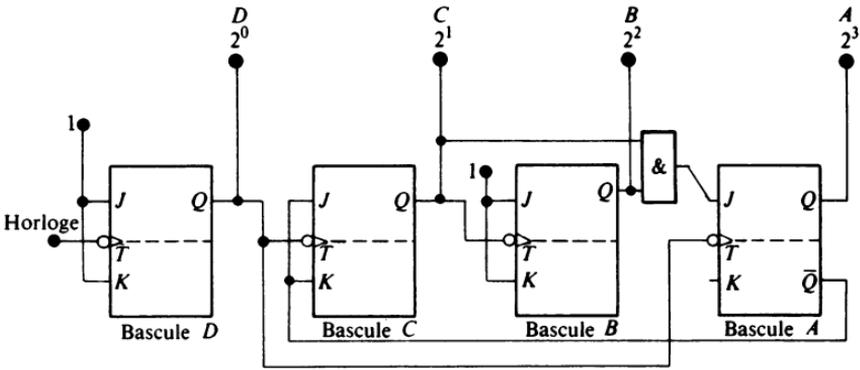


Fig. 22 Compteur asynchrone BCD 8421.

Les bascules B et D sont câblées en diviseur par deux. Au début du comptage, lorsque toutes les bascules sont à zéro, la sortie \overline{Q}_A est à 1, et comme elle alimente les entrées J et K de la bascule C, celle-ci fonctionne également en diviseur par deux. La porte ET branchée sur les sorties Q_B et Q_C délivre un « 0 » à l'entrée J de la bascule A, qui maintient ainsi sa sortie à 0. Le compteur compte alors normalement de 0 à 7. Après la septième impulsion, la sortie de la porte ET délivre un « 1 » à l'entrée J de la bascule A, car Q_B et Q_C sont à 1. A la huitième impulsion les trois bascules B, C et D reviennent à zéro. Cependant, la transition descendante de Q_D servant d'horloge à la bascule A, celle-ci enregistre le « 1 » présent en sortie de la porte ET avant qu'il ne disparaisse. La sortie Q_A est alors à « 1 », et \overline{Q}_A est à zéro. Comme la sortie \overline{Q}_A alimente les entrées J et K de la bascule C, celle-ci est maintenant bloquée en l'état, c'est-à-dire à zéro. La neuvième impulsion fait basculer Q_D à 1. La dixième impulsion fait revenir Q_D à zéro, et cette transition descendante servant d'horloge à la bascule A, celle-ci enregistre le « 0 » présent en sortie de la porte ET. Toutes les bascules sont revenues à l'état initial.

Compteur en anneau et compteur Johnson

Le compteur en anneau est un registre à décalage bouclé sur lui-même, tel que le montre la figure 23.

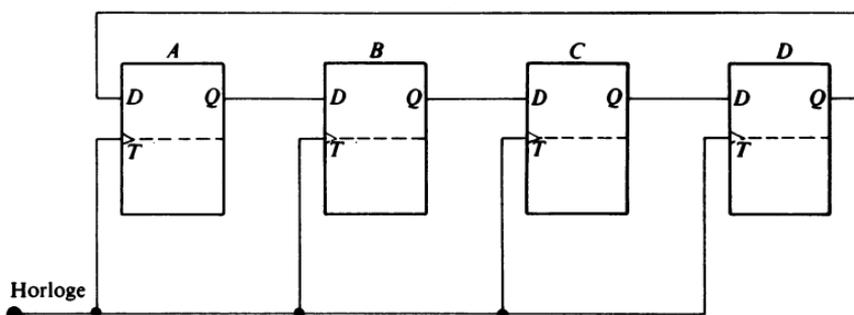


Fig. 23 Compteur en anneau.

Si toutes les bascules sont initialement à l'état « 0 », et que l'on introduit un « 1 » dans le registre, on constate que ce « 1 » circule de bascule en bascule, au rythme des impulsions d'horloge. Lorsque ce « 1 » sort de la bascule D il est réinjecté dans la bascule A, et le cycle continue indéfiniment.

Le décodage d'un tel compteur en anneau est aisé, car Q_A est à « 1 » lorsque le compteur compte 0, Q_B est à « 1 » lorsque le compteur compte 1, Q_C est à « 1 » lorsque le compteur compte 2, etc. Dans l'exemple de la figure 23, le compteur en anneau est un modulo 4. Si l'on désire une décade, il faut employer dix bascules, ce qui n'est pas une solution très économique.

Il est possible de doubler le cycle de comptage d'un compteur en anneau, en réalimentant l'entrée du registre par la sortie \overline{Q}_D au lieu de la sortie Q_D précédente. On obtient alors un compteur

Décimal	Q_A	Q_B	Q_C	Q_D
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

Fig. 24 Table de vérité d'un compteur Johnson.

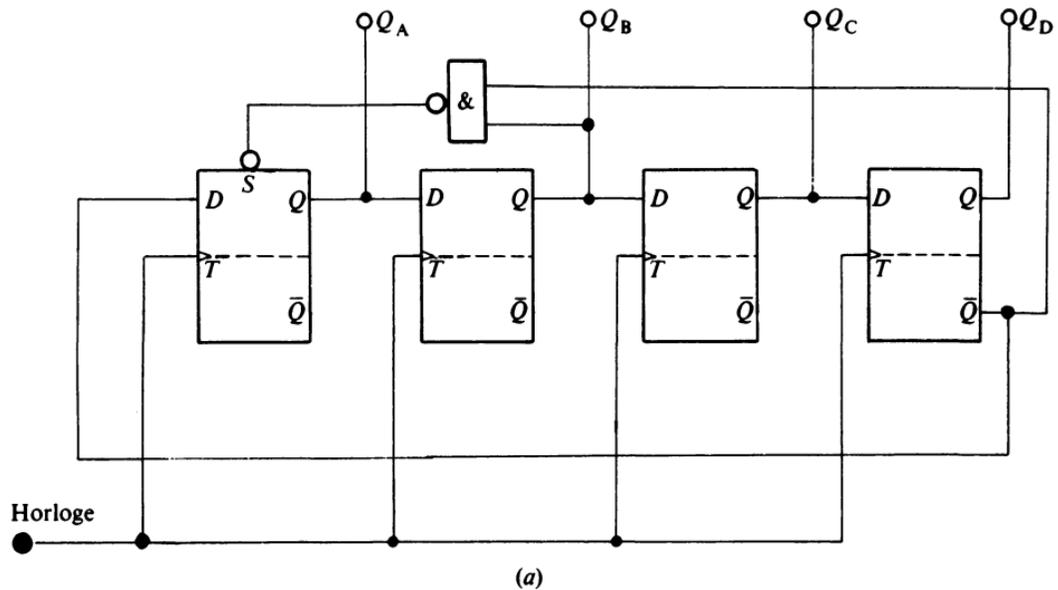


Fig. 25 Compteur Johnson avec remise en phase :
(a) réalisé avec des bascules D,

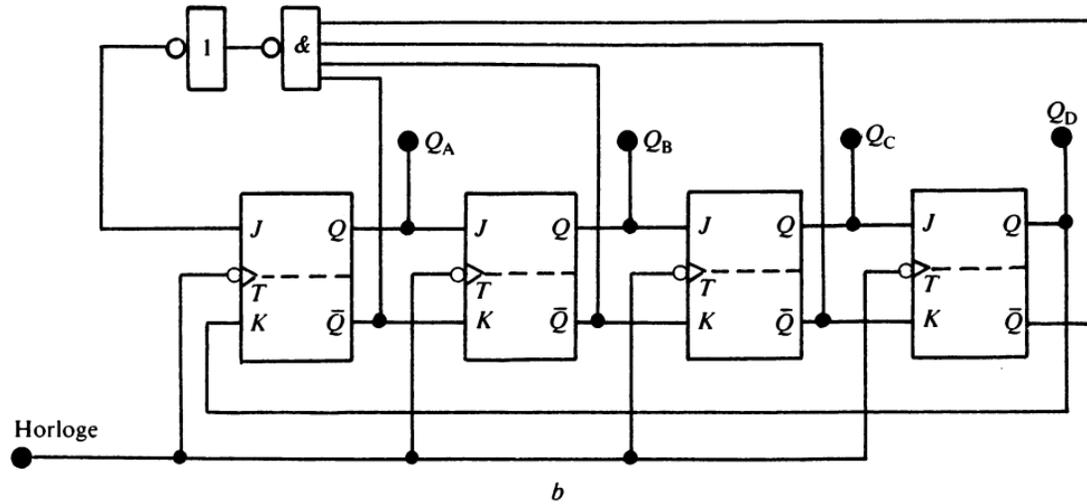


Fig. 25 Compteur Johnson avec remise en phase :
(b) réalisé avec des bascules JK.

Johnson. Un tel compteur est un peu plus difficile à décoder que le compteur en anneau, car il nécessite des portes OU-NON (NOR) à 2 entrées, mais il reste cependant plus facile à décoder que les compteurs classiques étudiés précédemment. La figure 24 donne la table de vérité d'un compteur Johnson modulo 8.

La remise en phase de ce compteur nécessite un circuit logique particulier, car à la mise sous tension, un tel compteur peut prendre une configuration parmi seize, autre que celle indiquée par la table de vérité. La figure 25 donne deux exemples de circuit de remise en phase d'un compteur Johnson.

- Circuit de remise en phase d'un compteur Johnson à bascules D (fig. 25 a). Ce circuit de remise en phase est constitué par une porte ET-NON dont les 2 entrées sont branchées sur les sorties Q_B et Q_D , et donc la sortie alimente l'entrée S de forçage de « 1 » de la bascule A . Lorsque la bascule D contient un 0 et que la bascule B contient un « 1 », la porte ET-NON force la bascule A à « 1 ». Cela dure ainsi jusqu'à ce que le registre soit rempli de « 1 ». La sortie de la porte ET-NON délivre alors un « 1 » qui est inactif pour l'entrée de forçage, et la bascule A peut alors fonctionner normalement.

- Circuit de remise en phase d'un compteur Johnson à bascules JK (fig. 26 b). Ce circuit de remise en phase est constitué par une porte ET-NON à 4 entrées branchées sur les sorties $\overline{Q_A}$, $\overline{Q_B}$, $\overline{Q_C}$ et $\overline{Q_D}$, et dont la sortie alimente l'entrée J de la bascule A . La sortie Q de la bascule D alimente l'entrée K de la bascule A . Lorsque le registre ne contient que des « 0 » la porte ET-NON suivie d'un inverseur fournit un « 1 » sur l'entrée J de la bascule A , et la sortie Q_D fournit un « 0 » sur l'entrée K de la bascule A . Celle-ci enregistre donc un « 1 », au coup d'horloge suivant. La porte ET-NON (NAND) suivie d'un inverseur fournit alors un « 0 » sur l'entrée J de la bascule A , et la sortie Q_D fournit un « 0 » sur l'entrée K de la bascule A . Celle-ci reste donc à « 1 » au coup d'horloge suivant. Il en est ainsi jusqu'à ce que le registre soit rempli de « 1 ». L'entrée J étant toujours à 0, la sortie Q_D délivre alors un « 0 » à l'entrée K de la bascule A . Celle-ci passe à « 0 » au coup d'horloge suivant. Il en est ainsi jusqu'à ce que le registre soit rempli de « 0 ».

MONTAGES PRATIQUES

Le compteur Johnson à bascules D (modulo 8)

- Réaliser le compteur Johnson à bascules D de la figure 26, en utilisant 2 circuits SN 7474 (double bascule D) et 1/4 de circuit SN 7400 (quadruple porte ET-NON (NAND), à 2 entrées).

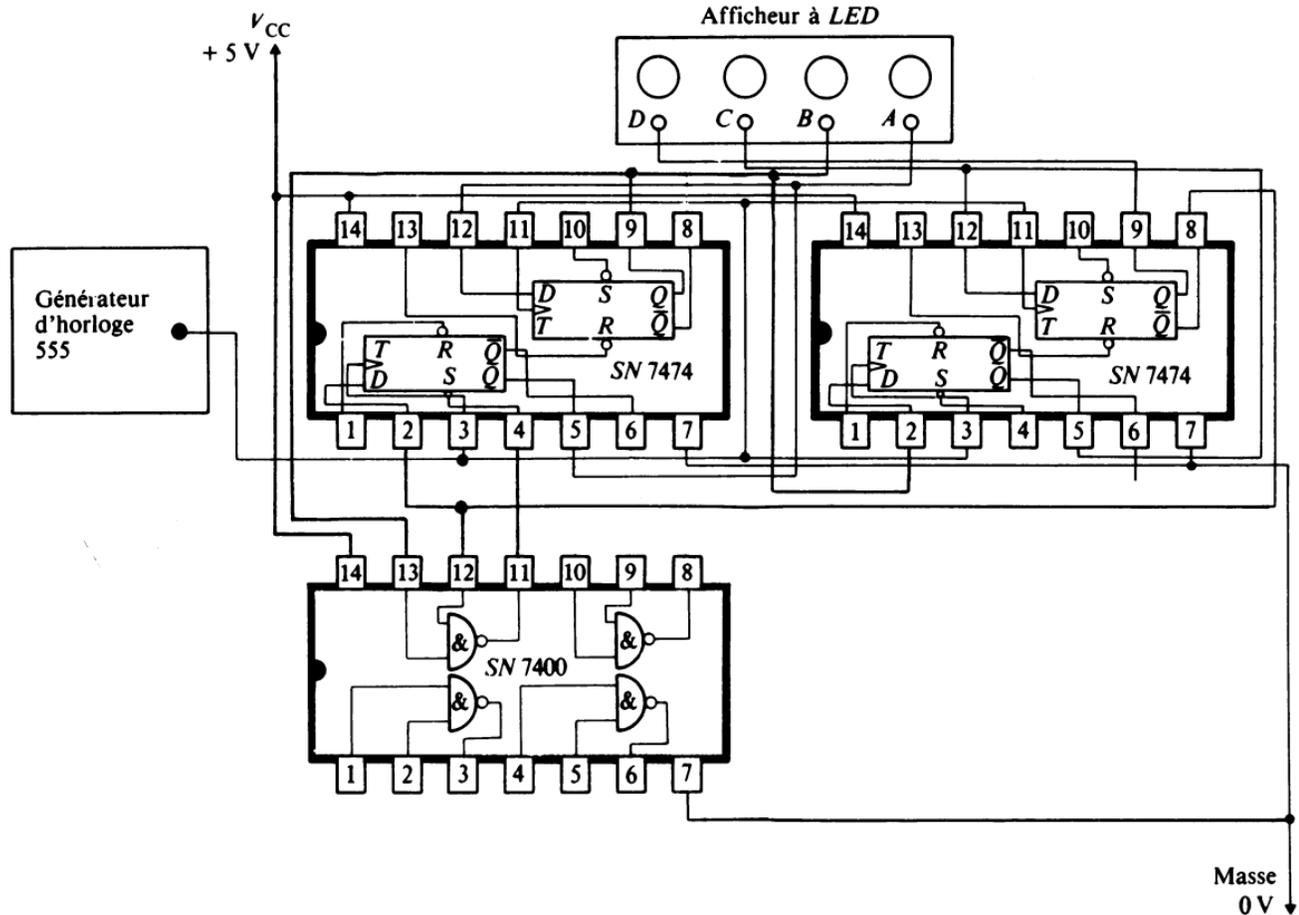


Fig. 26 Compteur Johnson à bascules D (modulo 8).

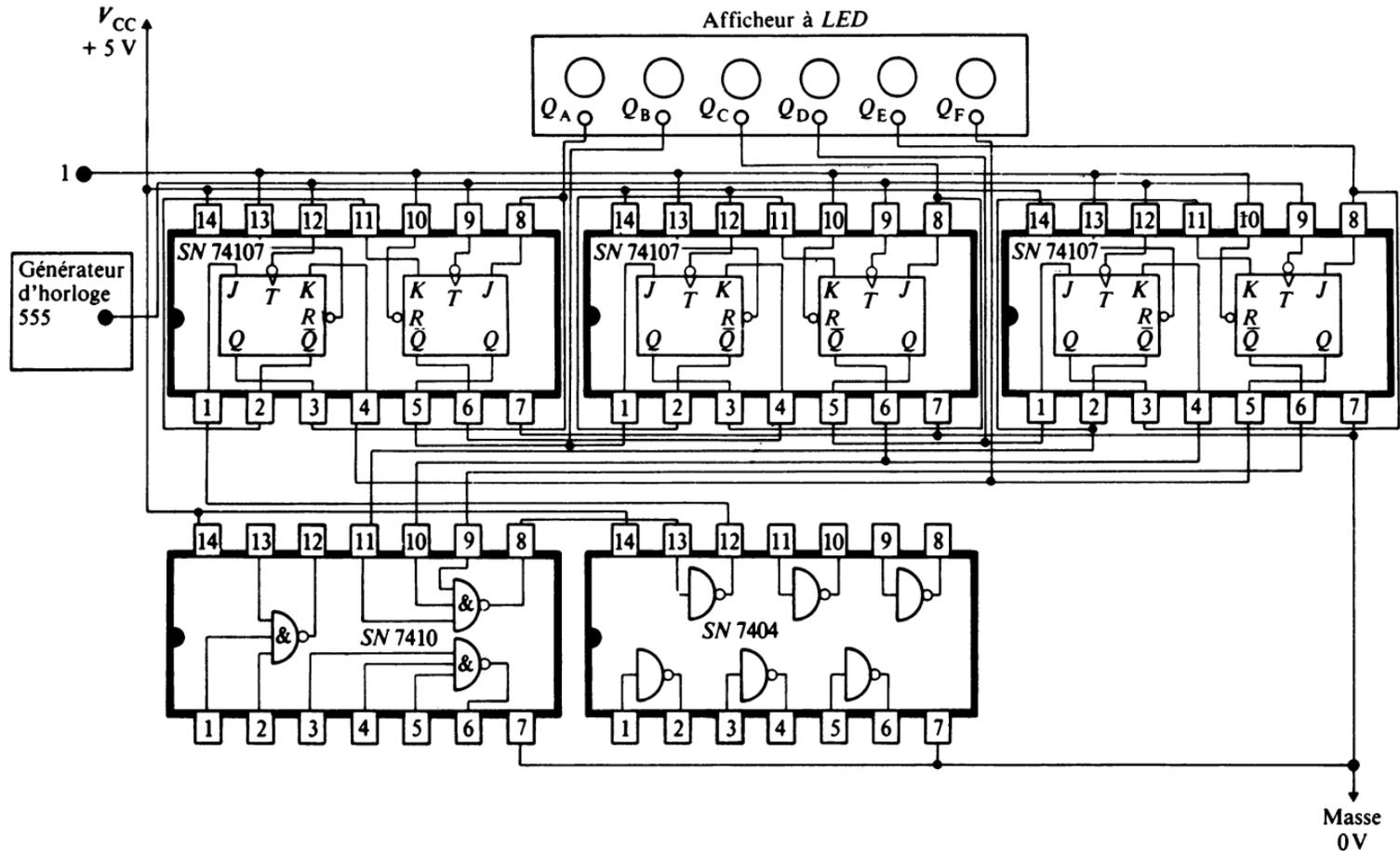


Fig. 27 Compteur Johnson à cycle pair (modulo 12).

- Appliquer un train d'impulsions et observer l'état des sorties Q des bascules. Dresser la table de vérité de ce compteur pour une séquence complète de 8 impulsions.

Le compteur Johnson à cycle pair (modulo 12)

- Réaliser le montage du compteur Johnson à cycle pair de la figure 27, en utilisant 3 circuits SN 74107 (double bascule JK), 1/3 de circuit SN 7410 (triple porte ET-NON à 3 entrées) et 1/6 de circuit SN 7404 (sextuple inverseur).

Décimal	Q_A	Q_B	Q_C	Q_D	Q_E	Q_F
0	0	0	0	0	0	0
1	1	0	0	0	0	0
2	1	1	0	0	0	0
3	1	1	1	0	0	0
4	1	1	1	1	0	0
5	1	1	1	1	1	0
6	1	1	1	1	1	1
7	0	1	1	1	1	1
8	0	0	1	1	1	1
9	0	0	0	1	1	1
10	0	0	0	0	1	1
11	0	0	0	0	0	1

Fig. 28 Table de vérité d'un compteur Johnson à cycle pair (modulo 12).

- Appliquer un train d'impulsions et observer l'état des sorties Q des bascules. Dresser la table de vérité de ce compteur pour une séquence complète de 12 impulsions, et la comparer à celle de la figure 28.

Le compteur Johnson à cycle impair (modulo 11)

On peut modifier un compteur Johnson à cycle pair en un compteur Johnson à cycle impair, en utilisant un circuit logique de décodage qui évite la configuration où toutes les bascules sont à « 1 ». Sur la table de vérité de la figure 28, on peut transformer le compteur modulo 12 en un compteur modulo 11, en évitant le 6 décimal, pour lequel toutes les bascules sont à « 1 ».

- Exécuter le montage du compteur Johnson à cycle impair de la figure 29, en utilisant 3 circuits SN 74107 (double bascule JK), 1 circuit SN 7430 (porte ET-NON à 8 entrées), 2/3 de circuit SN

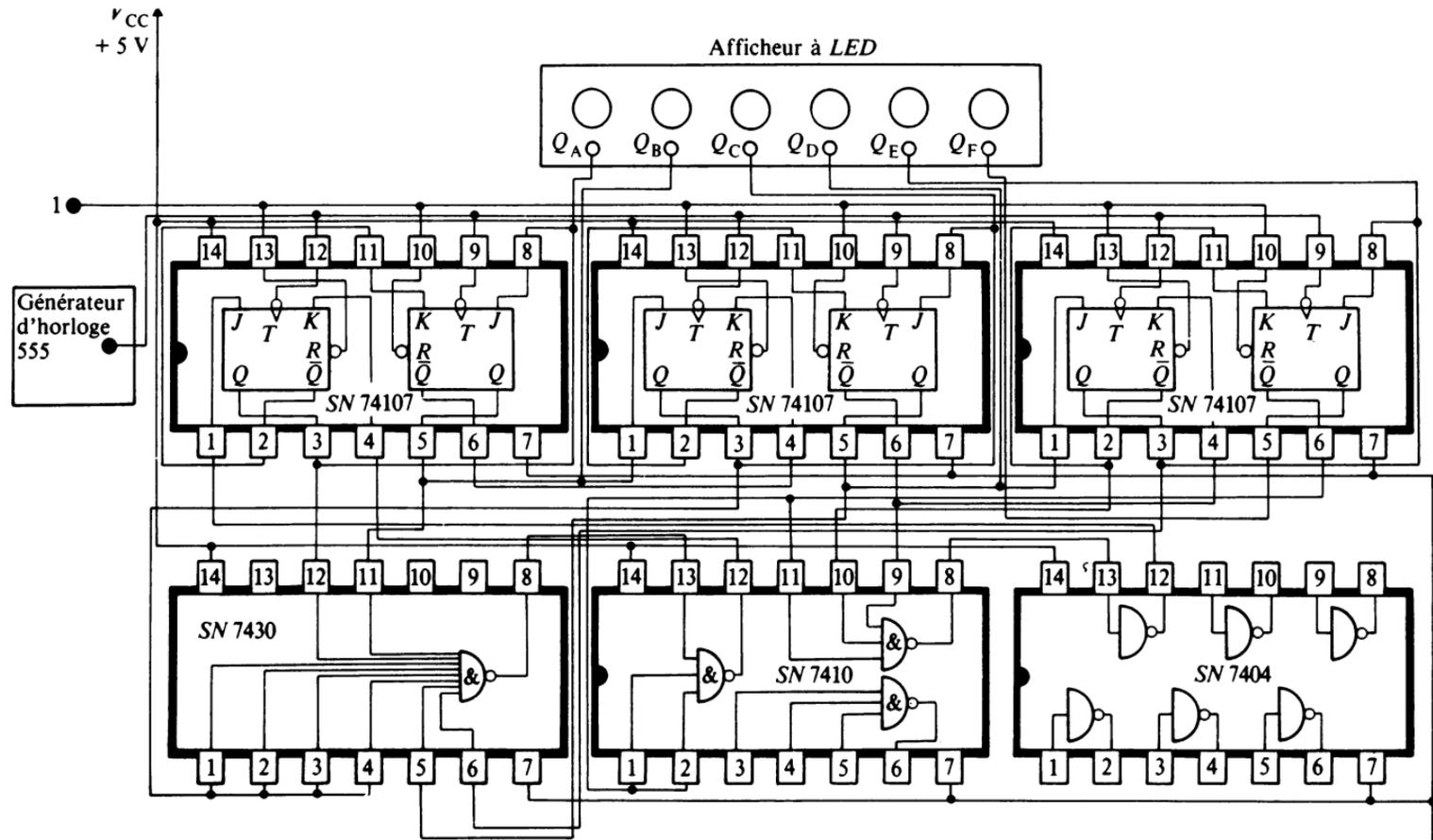


Fig. 29 Compteur Johnson à cycle impair (modulo 11).

7410 (triple porte ET-NON à 3 entrées) et 1/6 de circuit SN 7404 (sextuple inverseur).

- Appliquer un train d'impulsions et observer l'état des sorties Q des bascules. Dresser la table de vérité de ce compteur pour une séquence complète de 11 impulsions.

Générateurs pseudo-aléatoires

Un générateur pseudo-aléatoire est un registre à décalage qui fabrique un code dont les éléments binaires ne suivent aucun ordre particulier, c'est-à-dire qu'ils sont produits de manière aléatoire. Ceci s'obtient généralement avec un registre à décalage dont l'entrée est alimentée par un circuit logique plus ou moins complexe, lui-même alimenté par les bascules du registre.

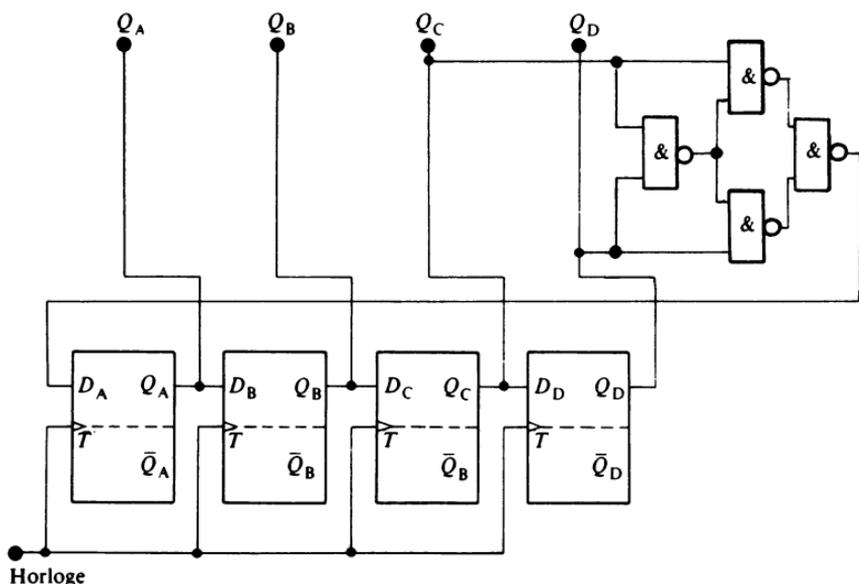


Fig. 30 Générateur pseudo-aléatoire.

La figure 30 donne un exemple simple de générateur *pseudo-aléatoire* fournissant une séquence qui n'est pas purement aléatoire, puisque toutes les combinaisons sauf une du registre sont explorées en un cycle complet, et qu'il est possible de prédire l'ordre des combinaisons suivant une table de vérité.

La porte OU-exclusif constituée par les 4 portes ET-NON sert à comparer l'état des sorties Q_C et Q_D des bascules C et D. Elle délivre à l'entrée D_A de la bascule A un « 1 » logique, lorsque Q_C et Q_D sont identiques, tel que l'indique la table de vérité de la figure 31. On voit que la même séquence se répète toutes les 15 impulsions.

Décimal	Q_A	Q_B	Q_C	Q_D
0	1	1	1	1
1	0	1	1	1
2	0	0	1	1
3	0	0	0	1
4	1	0	0	0
5	0	1	0	0
6	0	0	1	0
7	1	0	0	1
8	1	1	0	0
9	0	1	1	0
10	1	0	1	1
11	0	1	0	1
12	1	0	1	0
13	1	1	0	1
14	1	1	1	0

Fig. 31 Table de vérité d'un générateur pseudo-aléatoire.

MONTAGES PRATIQUES

Un générateur pseudo-aléatoire à 15 combinaisons

- Monter le générateur pseudo-aléatoire à 15 combinaisons de la figure 32, en utilisant 2 circuits SN 7474 de double bascule D et 1 circuit SN 7400 (quadruple porte ET-NON à 2 entrées).
- Appliquer un train d'impulsions et observer l'état des sorties Q des bascules. Dresser la table de vérité et vérifier qu'elle correspond à celle de la figure 31.

Une mémoire adressable

Il est possible d'utiliser le circuit SN 7475 (quadruple bascule D) comme une mémoire adressable à 4 bits (1/2 octet) en les associant et en les adressant à l'aide d'un décodeur BCD — décimal SN 7442, tel que l'indique la figure 33.

- Appliquer un quartet (4 bits) au registre d'entrée et l'adresse de la mémoire voulue au décodeur (SN 7442). Fournir un « 0 » logique à l'entrée de validation, et l'horloge du générateur d'horloge 555, et vérifier la mémorisation du quartet dans la mémoire adressée.

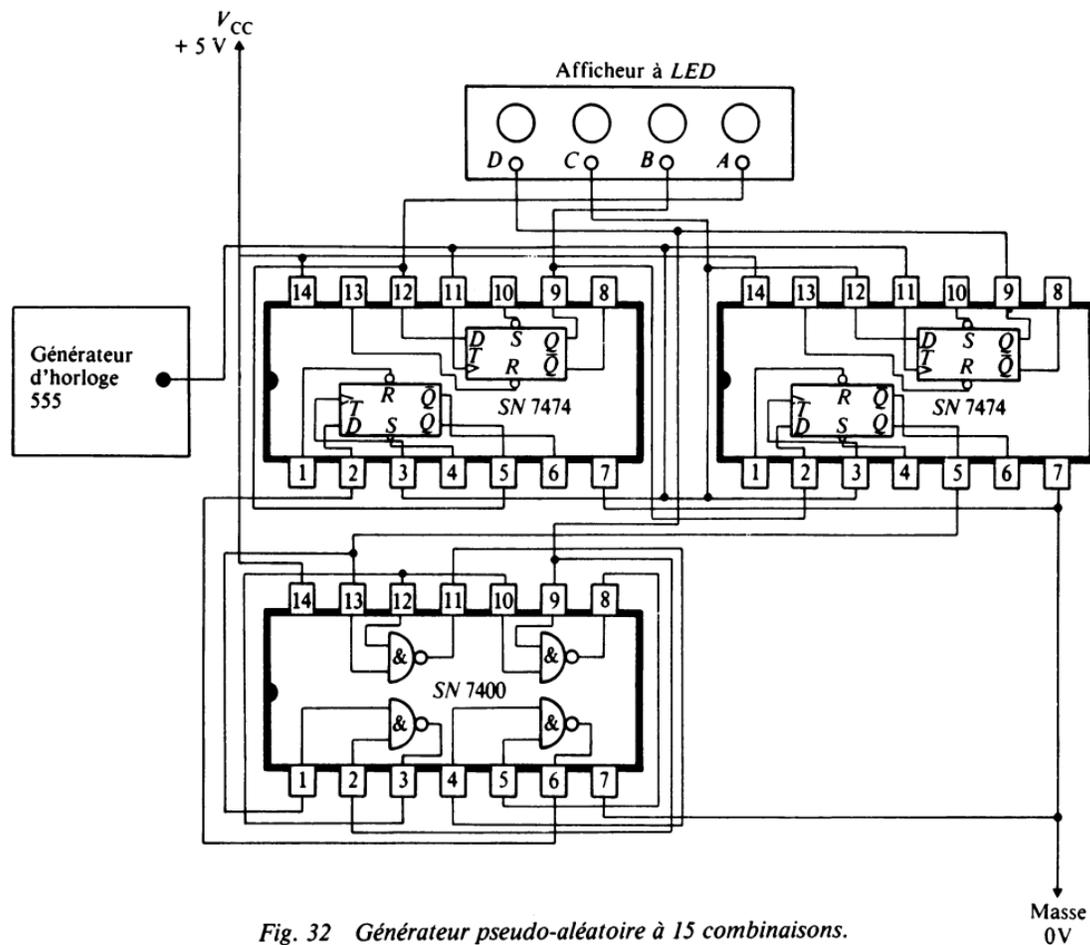


Fig. 32 Générateur pseudo-aléatoire à 15 combinaisons.

Masse
0V

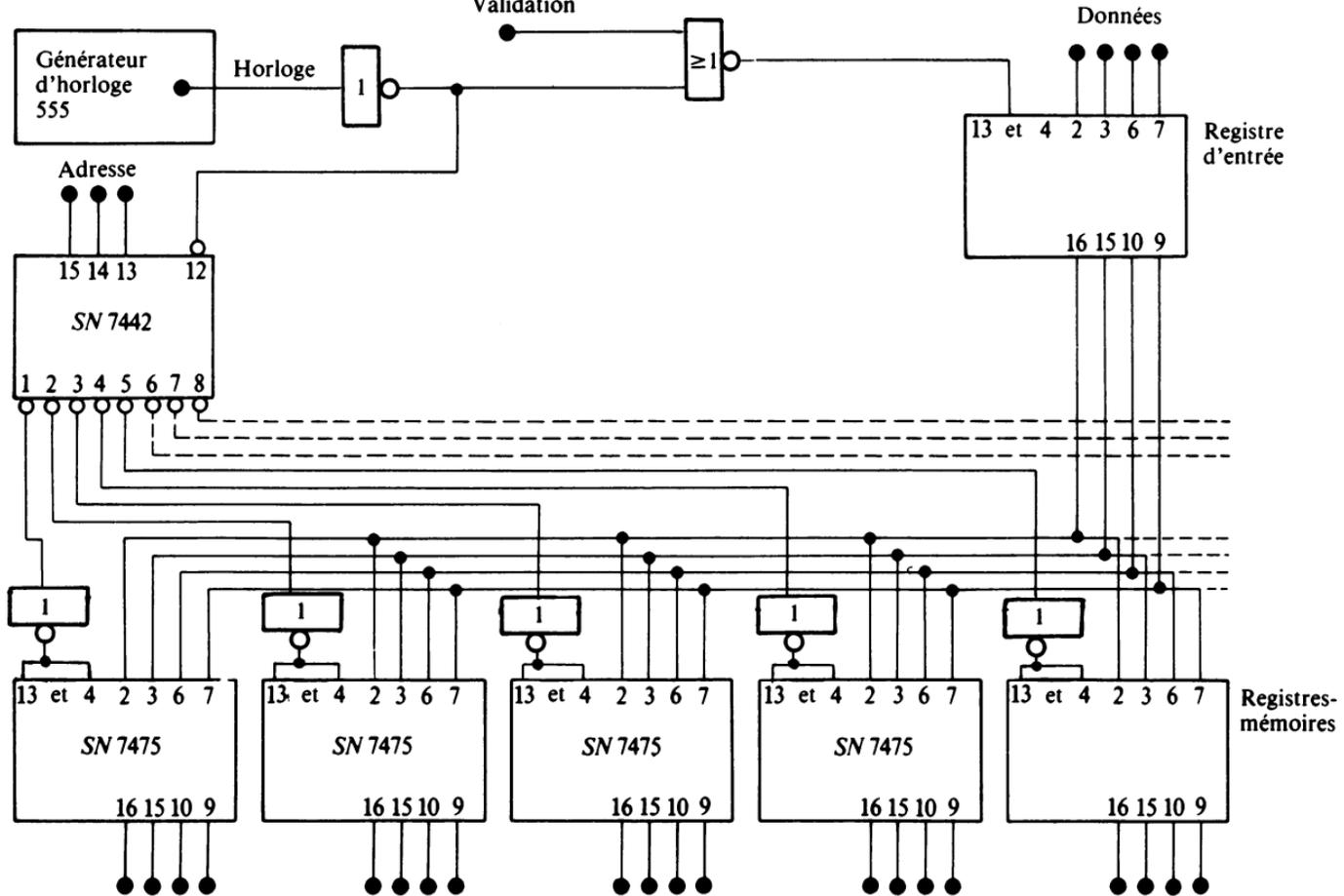


Fig. 33 Mémoire adressable.

- Répéter cette procédure en mémorisant d'autres quartets dans d'autres mémoires, en changeant d'adresse.

Un compteur asynchrone modulo 6

Il est possible de modifier une décade SN 7490, grâce à un circuit logique de décodage. Le signal de décodage est envoyé sur les entrées de remise à zéro, afin de raccourcir le cycle de comptage. Dans l'exemple de la figure 34, on applique les sorties Q_B et Q_C aux entrées de remise à zéro ($R_0(1)$ et $R_0(2)$), de façon à raccourcir le cycle de comptage au modulo 6.

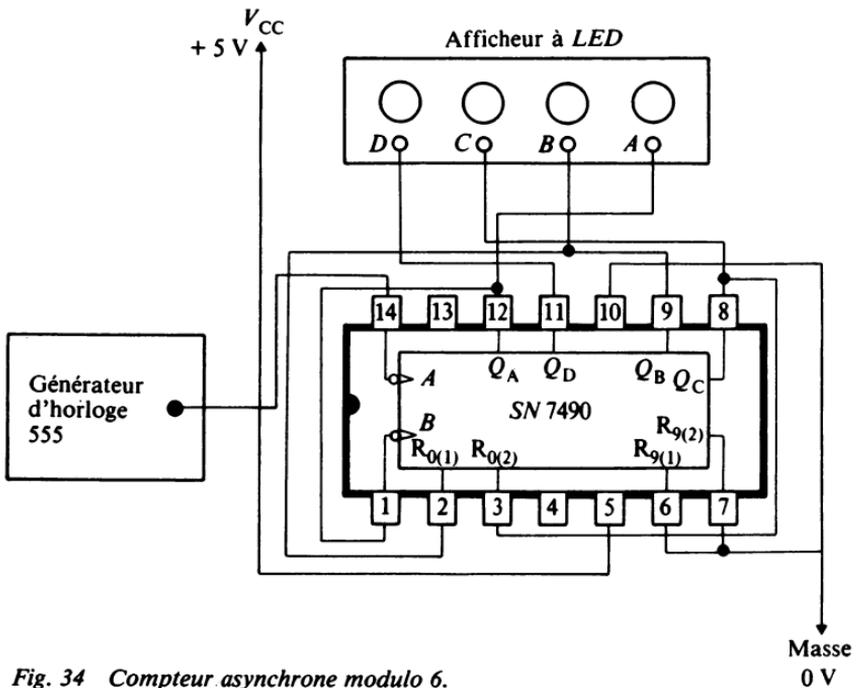


Fig. 34 Compteur asynchrone modulo 6.

- Appliquer un train d'impulsions, et observer l'état des sorties Q des bascules pour une séquence de 6 impulsions.

Un compteur asynchrone modulo 73

Il est possible de réaliser des compteurs à grand nombre d'états en associant deux décades SN 7490 suivant un code BCD.

- Réaliser le compteur asynchrone modulo 73 de la figure 35, en utilisant 2 circuits SN 7490 (décade asynchrone), 1/2 circuit SN 7420 (double porte ET-NON à 4 entrées) et 1/6 de circuit SN 7404 (sextuple inverseur).
- Appliquer un train d'impulsions, et observer l'état des sorties Q des bascules pour une séquence de 73 impulsions.

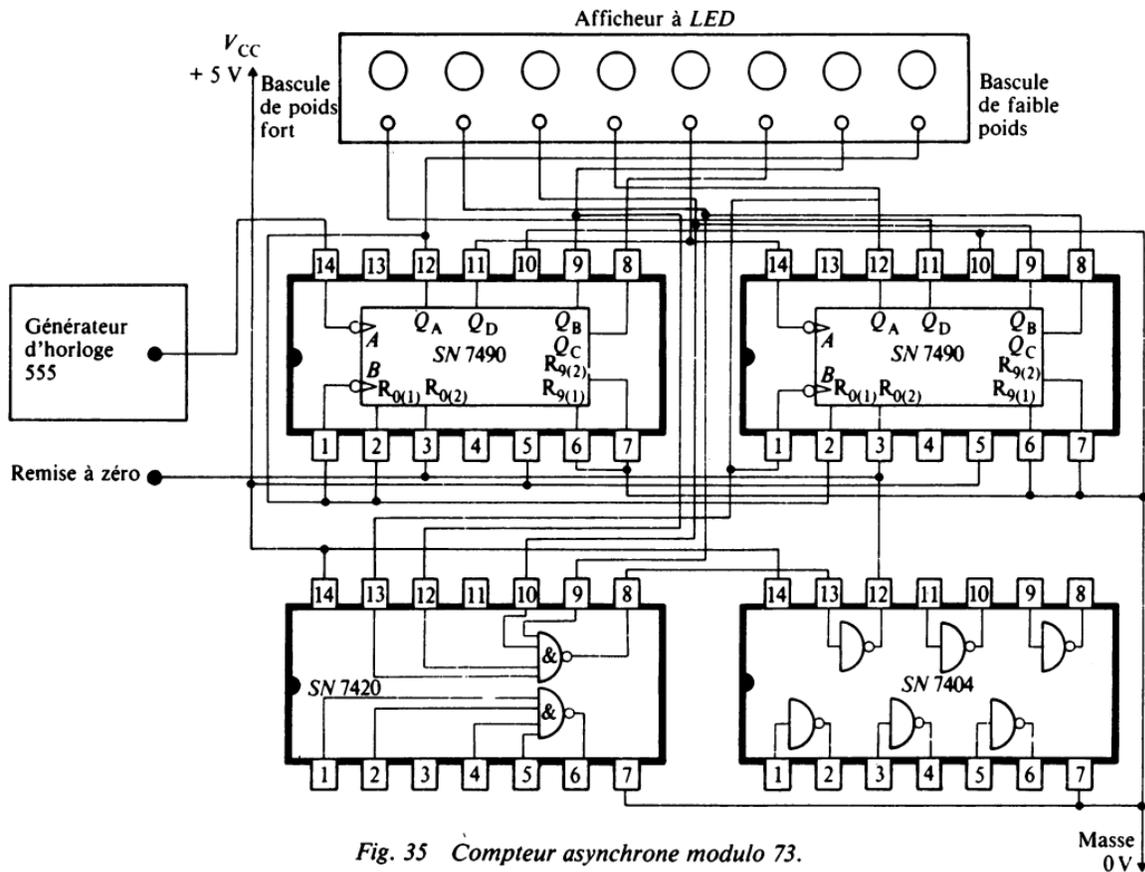


Fig. 35 Compteur asynchrone modulo 73.

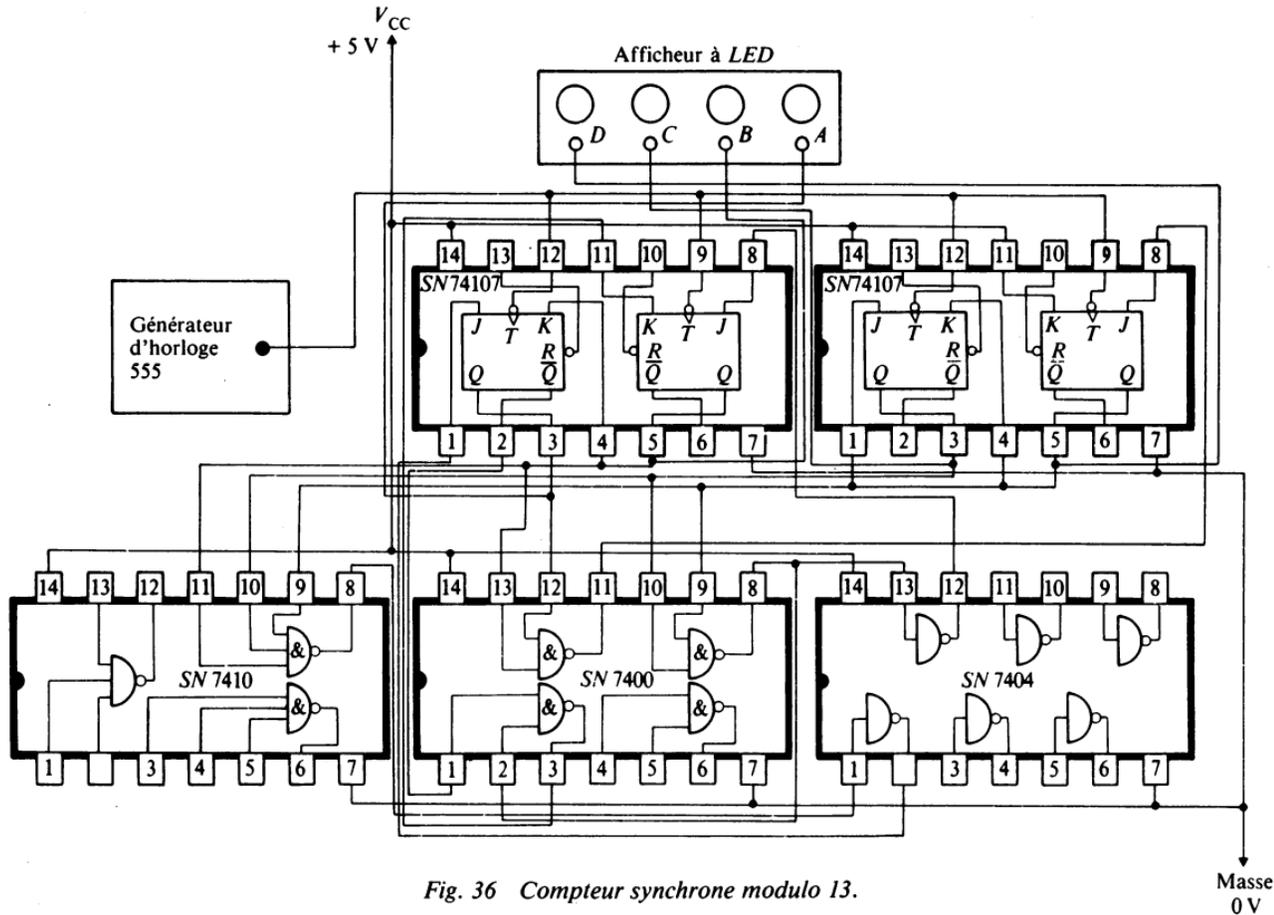


Fig. 36 Compteur synchrone modulo 13.

Un compteur synchrone modulo 13

- *Réaliser le montage de compteur synchrone modulo 13 de la figure 36, en utilisant 2 circuits SN 74107 (double bascule JK), 1/3 de circuits SN 7410 (triple porte ET-NON à 3 entrées), 3/4 de circuits SN 7400 (quadruple porte ET-NON à 2 entrées) et 1/3 de circuits SN 7404 (sextuple inverseur).*
- *Appliquer un train d'impulsions, et observer l'état des sorties Q des bascules pour une séquence de 13 impulsions.*

histoire de logique

quelques applications

Il existe une grande diversité d'application des systèmes logiques de commande. La plupart d'entre eux n'autorise le fonctionnement d'une machine ou d'un dispositif, qu'à condition que certains événements se produisent suivant un ordre préétabli. Cela implique que ces systèmes logiques de commande utilisent des éléments de mémoire, des compteurs et des bases de temps, ainsi que des portes logiques.

Un exemple simple illustrant ce propos est donné à la figure 1. Il s'agit d'un verrou à combinaisons que l'on peut ouvrir à condition de manœuvrer trois boutons d'ouverture dans l'ordre *C*, *A*, *B*. Toute autre séquence doit déclencher une alarme (visuelle ou sonore). Le verrou peut être refermé et l'alarme coupée, en appuyant en même temps sur les trois boutons *A*, *B* et *C*.

Ce chapitre décrit un certain nombre d'applications utilisant la logique combinatoire et séquentielle, pour lesquels sont donnés les paramètres de fonctionnement et le circuit logique permettant de les réaliser. La réalisation pratique de tels circuits est recommandée, de façon à pouvoir vérifier concrètement les fonctionnements.

Ces circuits peuvent, bien entendu être transposés à bien d'autres applications selon le gré du lecteur. Leur but est d'illustrer l'application des informations réunies jusqu'à présent.

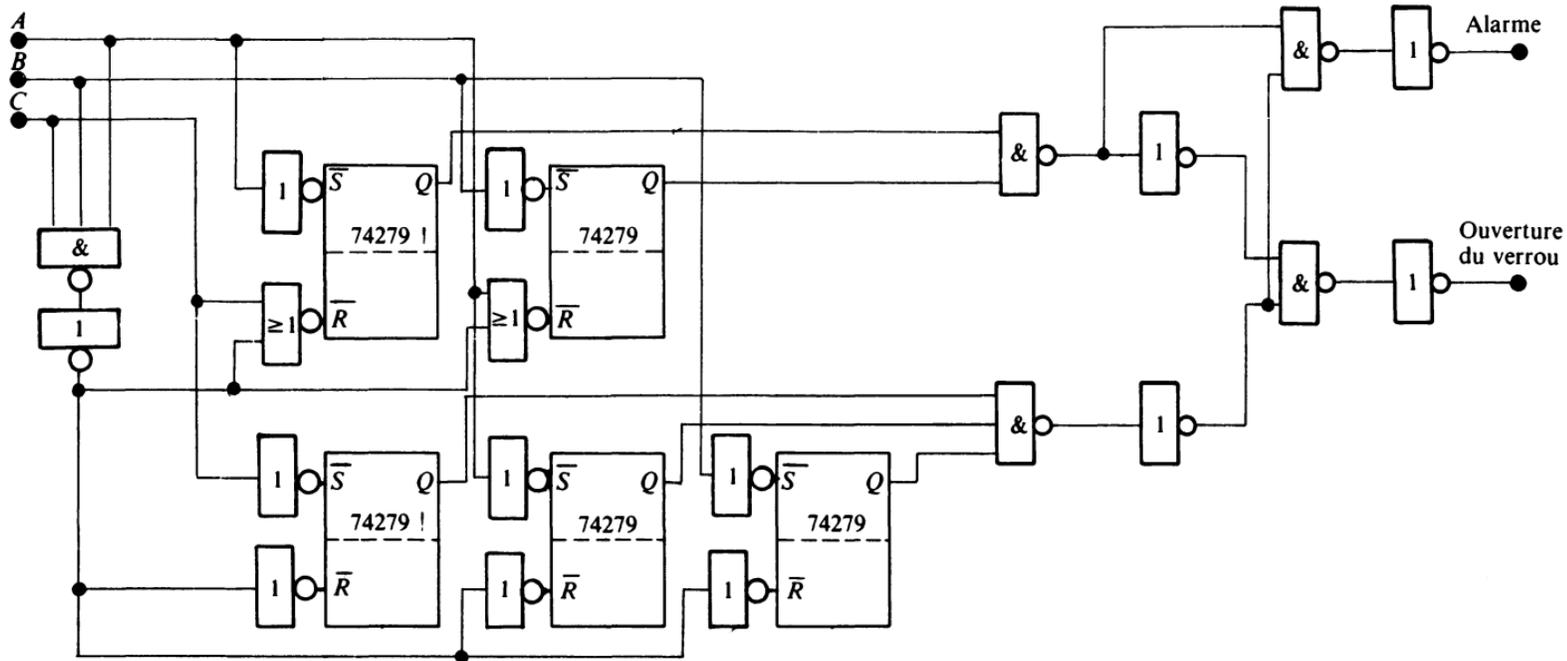


Fig. 1 Verrou à combinaisons.

Un système de sécurité

Les systèmes de sécurité sont utilisés dans de multiples domaines, que ce soit pour la protection d'une banque ou le contrôle d'explosifs.

Soit un magasin comprenant deux pièces. La première sert à stocker des équipements et la seconde des explosifs. Trois personnes peuvent avoir accès à ce magasin et possèdent chacune une clé différente. Il suffit de deux clés quelconques pour ouvrir la pièce aux provisions, alors qu'il faut les trois clés pour ouvrir la pièce aux explosifs.

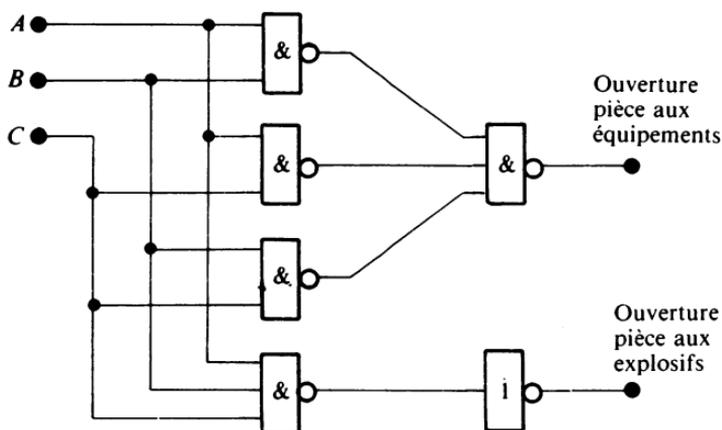


Fig. 2 Système de sécurité.

Le logigramme de la figure 2 répond à ces deux exigences.

- Le réaliser en utilisant 3/4 de circuit SN 7400 (quadruple porte ET-NON à 2 entrées), 2/3 de circuit SN 7410 (triple porte ET-NON à 3 entrées) et 1/6 de circuit SN 7404 (sextuple inverseur).
- Appliquer toutes les combinaisons de signaux aux entrées A, B, C et vérifier le fonctionnement de ce circuit en se servant de deux LED que l'on branche sur les sorties.

Contrôle de température

Dans un dispositif de contrôle de température. Un voyant vert doit s'allumer lorsque la température est normale; si la température devient trop élevée, le voyant vert doit s'éteindre et un voyant rouge doit s'allumer, en même temps qu'une alarme sonore doit se faire entendre. Un opérateur peut supprimer l'alarme sonore en appuyant sur un bouton, tout en maintenant l'alarme visuelle

rouge, jusqu'à la suppression du défaut. le logigramme de la figure 3 répond à ces conditions.

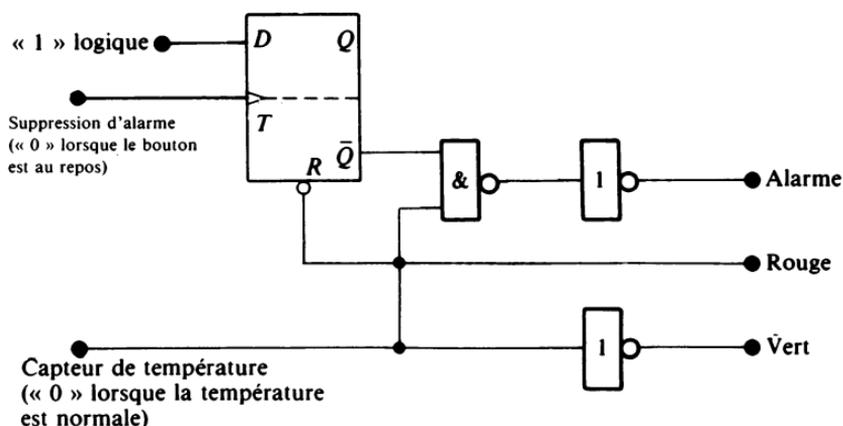


Fig. 3 Logigramme du contrôle de température.

- Réaliser ce montage (fig. 4), en utilisant : 1/2 de circuit SN 7474 (double bascule D). 1/4 de circuit SN 7400 (quadruple porte ET-NON à 2 entrées), 1/3 de circuit SN 7404 sextuple inverseur).
- Appliquer un niveau logique « 0 » à l'entrée « suppression d'alarme », et un niveau logique « 0 » à l'entrée « capteur de température ». Le voyant vert doit s'allumer.
- Appliquer un niveau logique « 1 » à l'entrée « capteur de température ». Le voyant vert doit s'éteindre, le voyant rouge doit s'allumer, ainsi que le voyant simulant l'alarme sonore.
- Appliquer un niveau logique « 1 » à l'entrée « suppression d'alarme ». Le voyant simulant l'alarme sonore doit s'éteindre, alors que le voyant rouge reste allumé.
- Appliquer un niveau logique « 0 » à l'entrée « capteur de température ». Le voyant vert doit s'allumer alors que le voyant rouge s'éteint.

Sonnette de service

Chaque chambre d'un hôtel est pourvue d'un bouton de sonnette permettant l'appel d'un employé. Lorsqu'un client appuie sur le bouton, cela a pour effet d'allumer un indicateur lumineux désignant le numéro de sa chambre. Si le client s'impatiente et qu'il appuie une seconde fois ou plusieurs fois de suite sur le bouton, une sonnerie se déclenche de façon à alerter l'employé de l'hôtel.

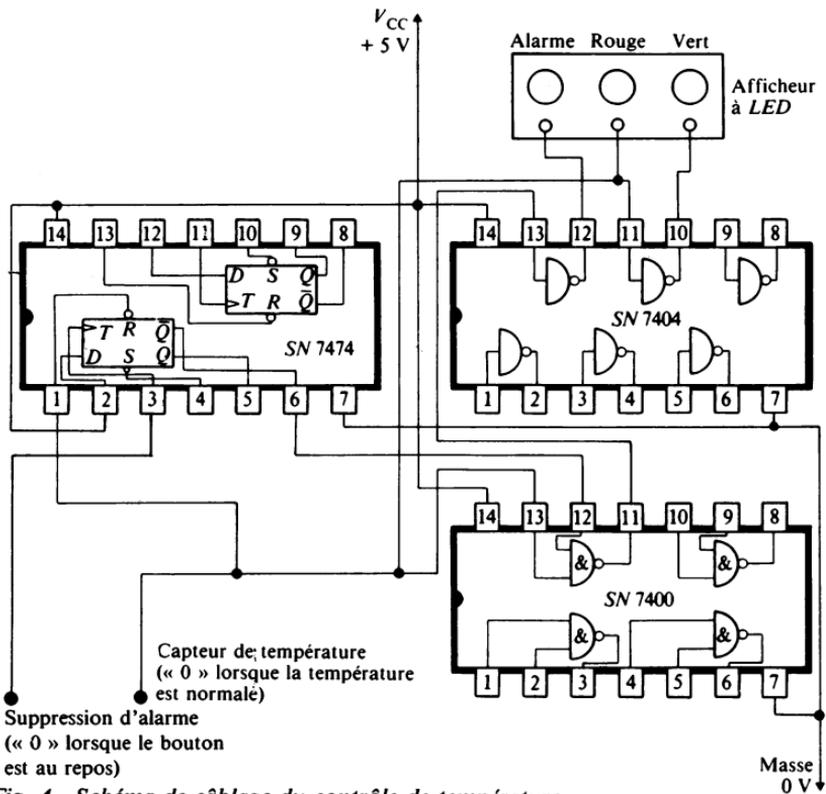


Fig. 4 Schéma de câblage du contrôle de température.

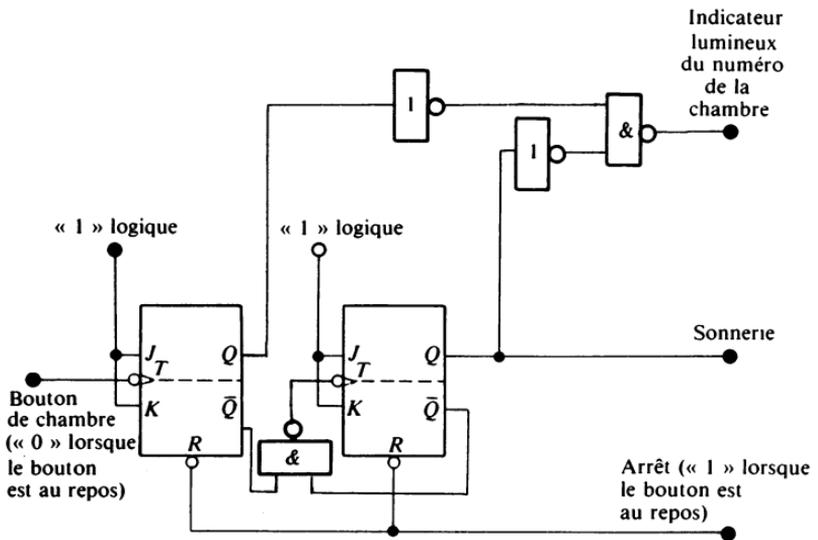


Fig. 5 Logigramme de la sonnette de service.

Un bouton permet à ce dernier d'éteindre l'indicateur lumineux et d'arrêter la sonnerie. Le logigramme de la figure 5 donne une solution au problème.

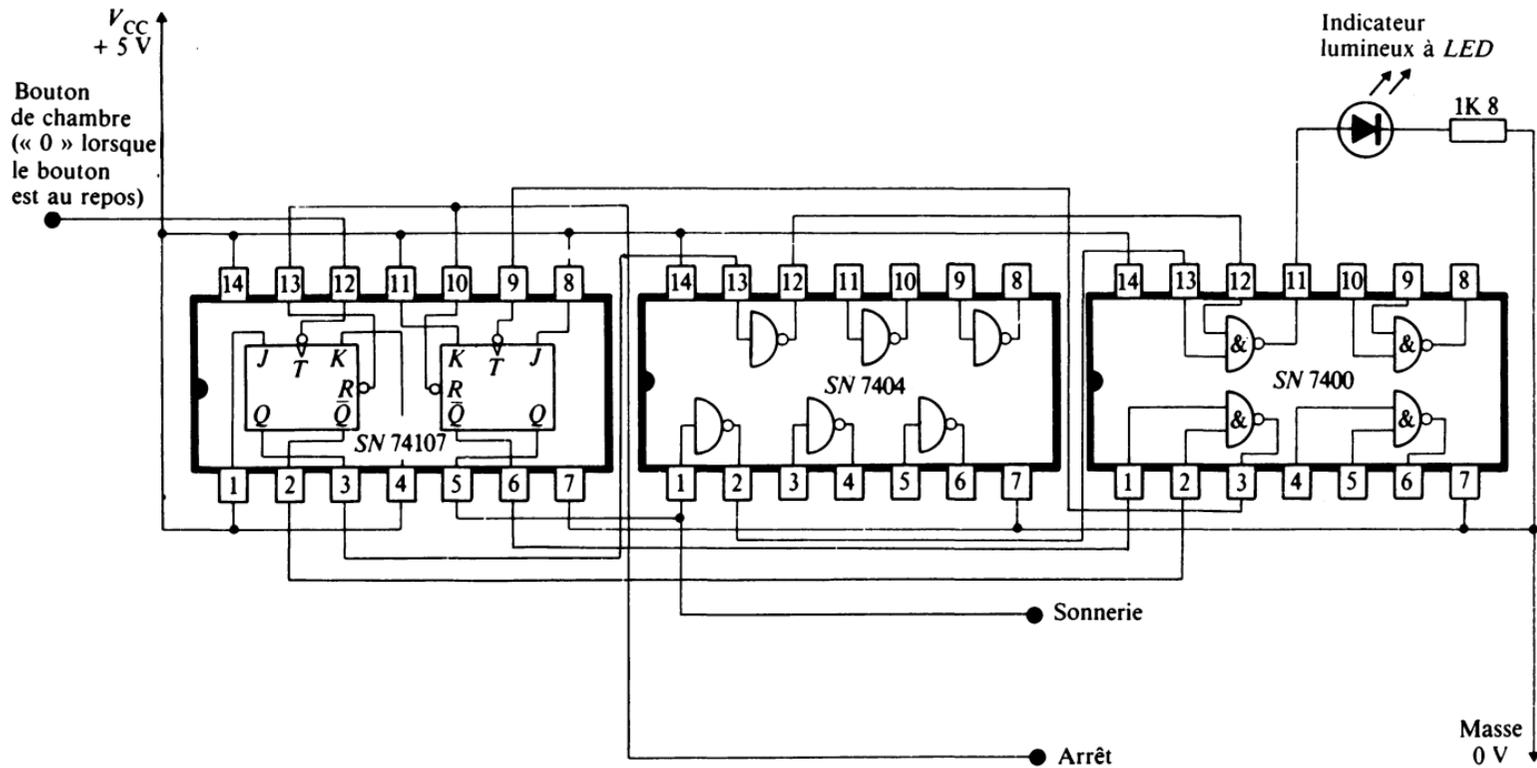


Fig. 6 Schéma de câblage de la sonnette de service.

- Câbler le montage de la figure 6, en utilisant : 1 circuit SN 74107 (double bascule JK), 1/2 circuit SN 7400 (quadruple porte ET-NON à 2 entrées), 1/3 de circuit SN 7404 (sextuple inverseur).
- Appliquer un niveau logique « 1 » pendant un bref instant à l'entrée « bouton de chambre ». La LED doit s'allumer.
- Appliquer une deuxième fois un niveau logique « 1 » pendant un bref instant à l'entrée « bouton de chambre ». La sonnerie doit se faire entendre et la LED doit rester allumée. Vérifier qu'en appliquant plusieurs fois de suite un niveau « 1 » sur cette entrée, le résultat demeure inchangé.
- Appliquer un niveau « 0 » sur l'entrée « arrêt ». La sonnerie doit s'arrêter et l'indicateur lumineux s'éteindre.

Simulation d'une commande d'ascenseur

Un ascenseur dessert deux étages. Chaque étage possède un bouton d'appel, et l'ascenseur possède deux boutons marqués 1 et 2. Lorsqu'une personne désire aller du 1^{er} au 2^e étage, par exemple, elle appuie sur le bouton d'appel de l'ascenseur, pénètre dans l'ascenseur, et appuie sur le bouton 2 de la cabine, pour faire monter l'ascenseur.

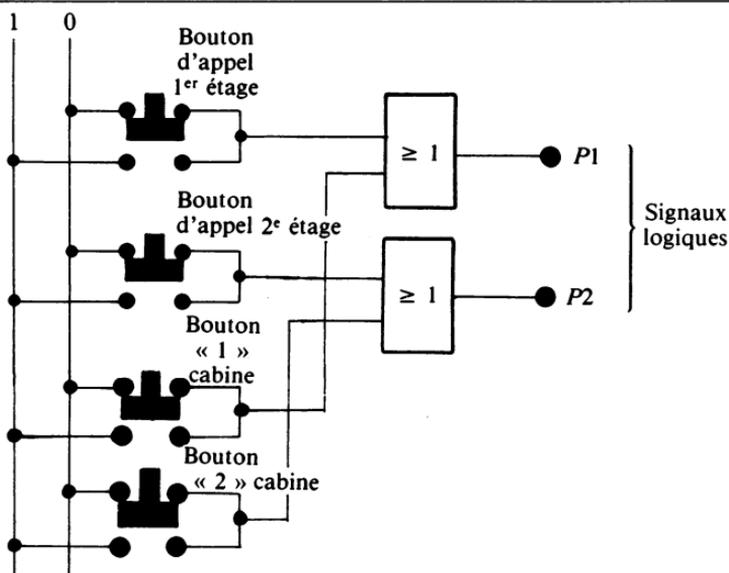


Fig. 7 Signaux logiques de commande de l'ascenseur :

- (a) boutons de commande,
 (b) signaux logiques

Le système logique de commande doit répondre aux conditions suivantes :

- on doit pouvoir appeler l'ascenseur de n'importe quel étage,
- on doit savoir lorsque l'ascenseur a atteint un étage,

- si l'ascenseur est en mouvement, le fait d'appuyer sur un bouton d'appel ne doit pas arrêter son mouvement,
- le moteur de l'ascenseur ne doit pas fonctionner, si par mégarde on appelle l'ascenseur, alors qu'il est déjà à l'étage.

La première condition est satisfaite en regroupant deux par deux le bouton d'appel du premier étage avec le bouton 1 de l'ascenseur, et le bouton d'appel du deuxième étage avec le bouton 2 de l'ascenseur, comme indiqué en figure 7.

La deuxième condition est satisfaite en plaçant un détecteur d'arrêt à chaque étage, c'est-à-dire un interrupteur qui bascule lorsque l'ascenseur atteint l'étage considéré. Chaque détecteur d'arrêt délivre un « 1 » logique lorsque l'ascenseur est à l'étage considéré, et il délivre un « 0 » logique lorsqu'il est en mouvement.

La troisième condition est satisfaite en commandant le moteur de l'ascenseur par deux bascules RS (fig. 8).

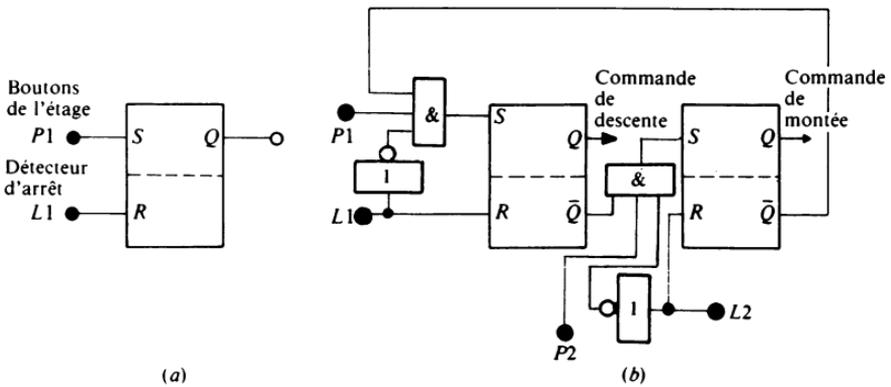


Fig. 8 Signaux logiques de commande du moteur :
 (a) d'un étage,
 (b) des deux étages.

L'ascenseur est supposé être au 1^{er} étage. Le détecteur $L1$ délivre un niveau logique « 1 » qui force la bascule de descente à « 0 ». Lorsque l'on appuie sur le bouton d'appel du deuxième étage, ou sur le bouton 2 de la cabine, le signal logique $P2$ vaut « 1 » et il force la bascule de montée à « 1 ». Le moteur est alors alimenté de façon à faire monter l'ascenseur. Lorsque celui-ci arrive au deuxième étage, le détecteur d'arrêt délivre un niveau logique « 1 » qui force la bascule de montée à « 0 »; le moteur s'arrête. L'ascenseur est maintenant prêt à redescendre.

La quatrième condition est satisfaite en interdisant la commande de descente du moteur lorsque l'ascenseur est au premier étage, et la commande de montée du moteur lorsque l'ascenseur est au deuxième étage. Le logigramme complet de cette commande d'ascenseur est donné à la figure 9.

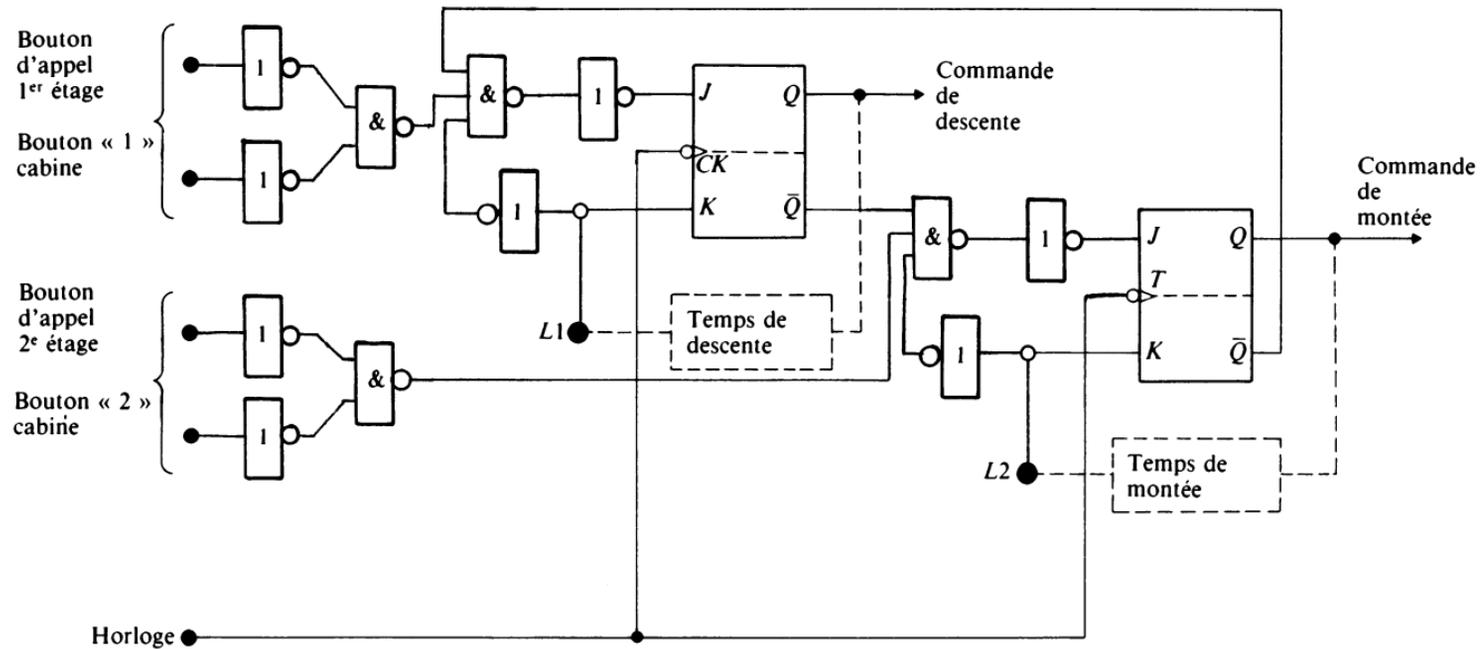


Fig. 9 Logigramme de la commande d'ascenseur.

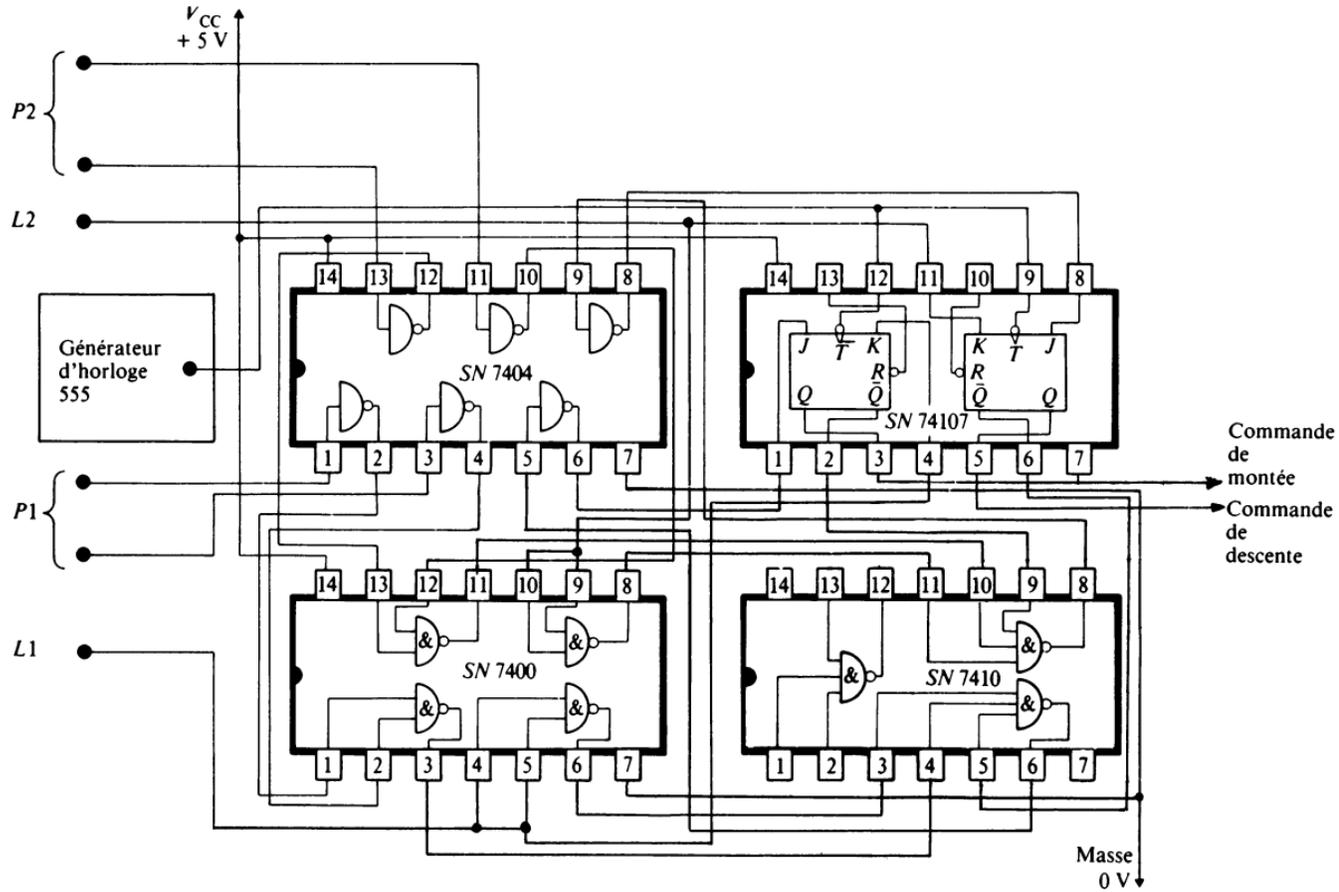


Fig. 10 Schéma de câblage de la commande d'ascenseur.

- Réaliser le montage de la figure 10, en utilisant : 1 circuit SN 74107 (double bascule JK), 1/2 circuit SN 7400 (quadruple porte ET-NON à 2 entrées), 2/3 de circuit SN 7410 (triple porte ET-NON à 3 entrées), 1 1/3 de circuit SN 7404 (sextuple inverseur).
- Régler la fréquence d'horloge du générateur d'horloge 555, à une valeur assez rapide. Appliquer les niveaux logiques correspondant aux commandes de l'ascenseur, et vérifier son fonctionnement avec deux LED branchées sur la sortie Q des bascules de montée et de descente. On simule l'arrêt automatique de l'ascenseur en appliquant le niveau logique correspondant au fonctionnement du détecteur, après quelques secondes de déplacement de la cabine.

Passage à niveau à commande automatique

On se propose de fermer automatiquement une barrière de passage à niveau en produisant un signal lumineux clignotant, lorsqu'un train approche de ce passage à niveau, et de l'ouvrir automatiquement en arrêtant le signal lumineux de signalisation, dès que le train a complètement dégagé le passage à niveau.

Les signaux de commande s'obtiennent à partir de trois détecteurs de position du train, placés en amont, au passage, et en aval du passage à niveau. Le détecteur amont est placé suffisamment loin avant le passage à niveau pour que l'on soit averti à temps de l'arrivée du train. Le détecteur aval est placé suffisamment loin après le passage à niveau pour que le train ait complètement dégagé ce passage avant l'ouverture de la barrière et l'arrêt de la signalisation (alarme).

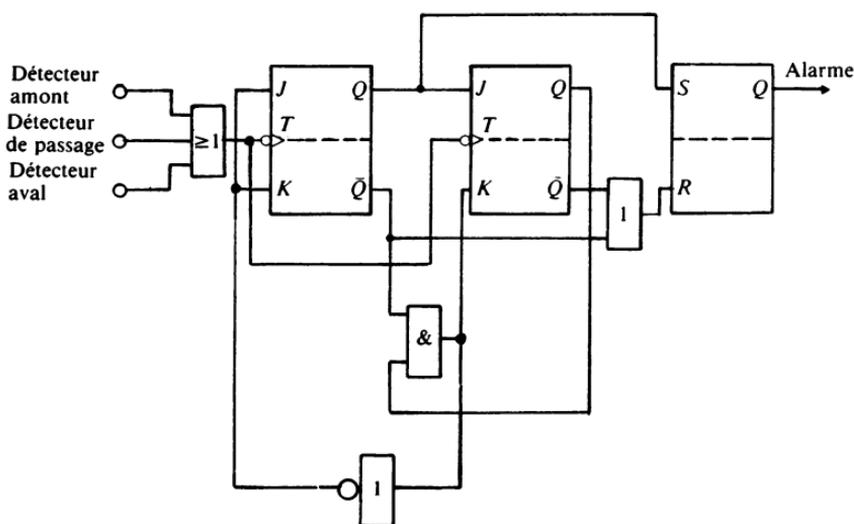


Fig. 11 Logigramme du passage à niveau à commande automatique.

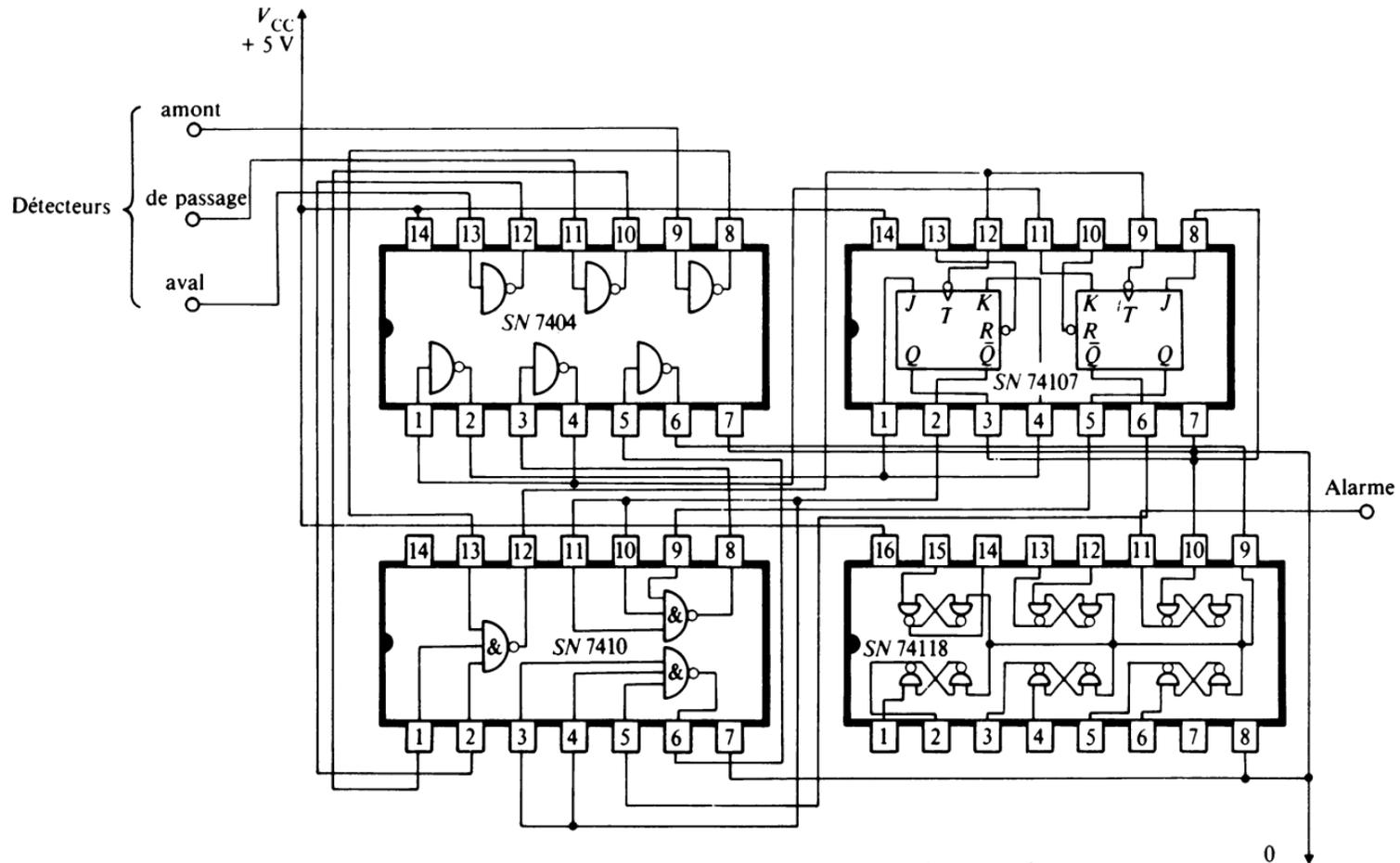


Fig. 12 Schéma de câblage du passage à niveau à commande automatique.

Les détecteurs sont conçus de manière à fournir un « 1 » logique lorsque le train passe au-dessus d'eux, et un niveau logique « 0 » lorsqu'ils sont au repos. Le logigramme de la figure 11 montre l'un des principes de réalisation de cette commande automatique. Il faudrait, en pratique, réaliser des contacts anti-rebonds (voir annexe) pour éviter des problèmes éventuels de fonctionnement (dus surtout aux étincelles sur les rails).

Les trois détecteurs sont appliqués à l'entrée d'une porte OU dont la sortie est reliée aux entrées d'horloge d'un compteur modulo 3 (c'est-à-dire d'un compteur binaire simple utilisant deux bascules JK). Ce compteur commande une bascule RS qui actionne l'alarme visuelle et sonore.

• Réaliser le montage de la figure 12 en utilisant : 1/6 de circuit SN 74118 (sextuple bascule RS), 1 circuit SN 7404 (quadruple inverseur), 1 circuit SN 7410 (triple ET-NON à 3 entrées), 1 circuit SN 74107 (double bascule JK).

• Appliquer un « 1 » à l'entrée « détecteur amont » et observer l'état de la commande de fermeture (alarme) en utilisant une diode LED.

• Appliquer un niveau « 1 » au « détecteur de passage » puis un niveau « 0 ». Vérifier que l'alarme subsiste.

• Enfin, appliquer un niveau logique « 1 » au « détecteur aval » puis un niveau « 0 ». Vérifier que l'alarme cesse dès que cette entrée est à « 1 ».

Passage protégé pour piétons

On considère un passage protégé pour piétons à commande manuelle. Lorsque l'on appuie sur le bouton, l'image d'un piéton rouge s'allume, indiquant qu'il ne faut pas traverser. Les feux de circulation précédemment verts, deviennent alors oranges pendant 5 s, puis passent au rouge. Simultanément, un piéton vert s'allume indiquant que l'on peut traverser. Au bout de 10 s, le piéton vert se met à clignoter, indiquant qu'on ne doit plus s'engager sur la chaussée. Le clignotement dure 5 s, puis le piéton rouge s'allume et les feux de circulation redeviennent verts. Le schéma logique de la figure 13 accomplit l'ensemble de ces opérations.

• Réaliser le montage de la figure 14, en utilisant : 4 circuits SN 74121 (monostable), 1 circuit SN 74279 (quadruple bascule RS), 1 1/2 circuit SN 7400 (quadruple porte ET-NON à 2 entrées), 1 circuit SN 7404 (sextuple inverseur).

• Régler la fréquence du générateur d'horloge 555 à 1 Hz. Appliquer un « 0 » logique sur l'entrée « bouton de commande », et observer l'état des voyants, en s'assurant que :

— Le piéton rouge et le feu orange s'allument pendant 5 s,

— le piéton vert et le feu rouge s'allument ensuite pendant 10 s,

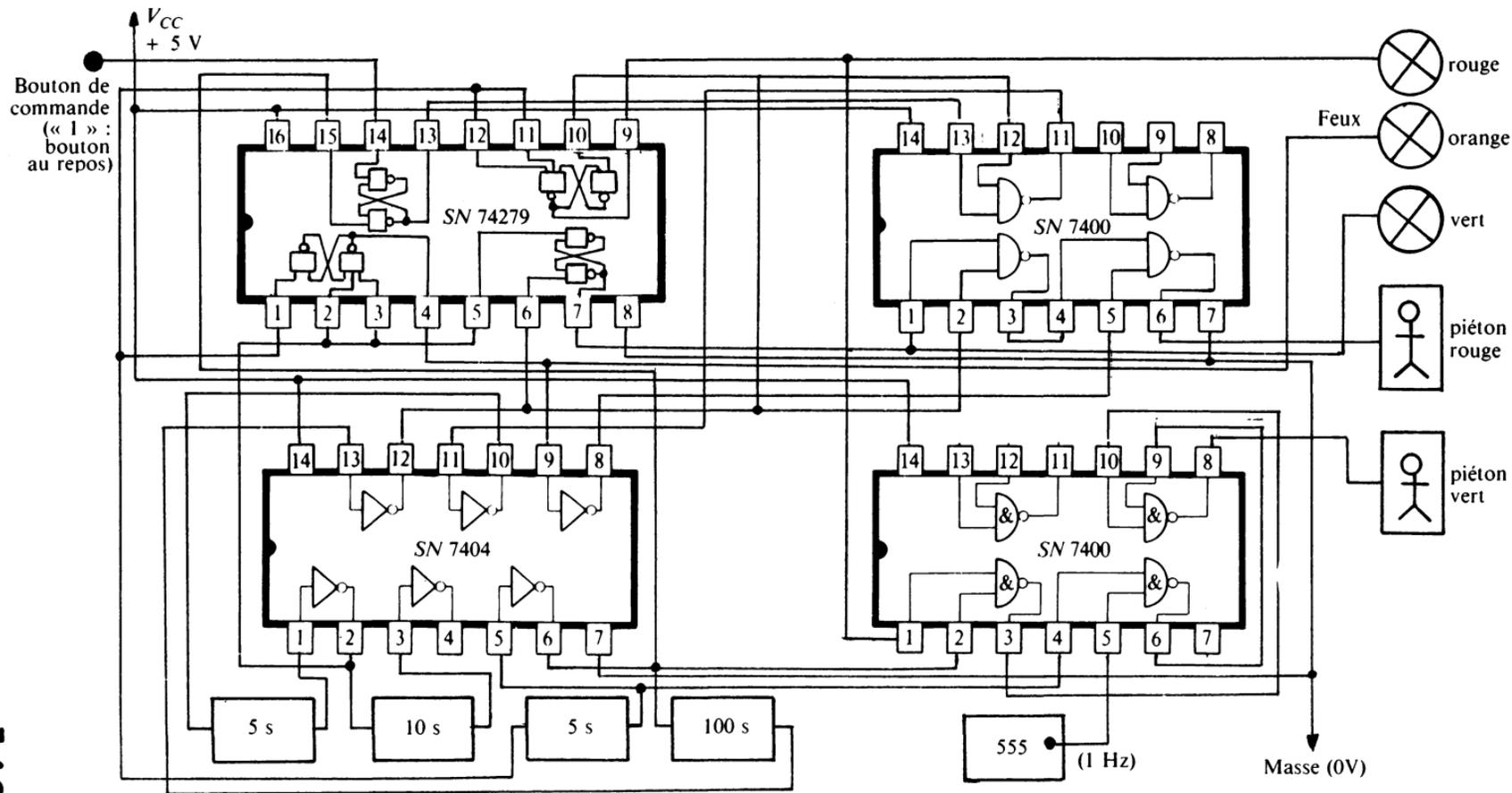


Fig. 14 Schéma de câblage du passage protégé pour piétons.

- le piéton vert clignotement ensuite pendant 5 s, le feu rouge restant allumé,
- le piéton rouge et le feu vert s'allument ensuite,
- le piéton rouge s'éteint au bout de 100 s.

REMARQUE : La temporisation de 100 s est indispensable pour que le trafic automobile puisse se faire, car les piétons pourraient bloquer le feu au rouge, en appuyant continuellement sur le bouton de commande.

des bêtes de somme

généralités sur les ordinateurs

Jadis, lorsqu'il était question d'ordinateur, les opinions des gens pouvaient se ranger en deux catégories. Il y avait ceux qui voyaient en la machine un nouveau Dieu sachant tout et pouvant tout, et les autres qui déclaraient que cette maudite machine ne fonctionnerait jamais et que de toute façon, elle ne remplacerait jamais un bon comptable avec un crayon et une gomme. Les premiers étaient un peu optimistes, les seconds trop rétrogrades. Il est bien évident que la vérité n'est détenue par aucun d'entre eux, qu'un ordinateur n'est qu'une machine, fabriquée par l'homme pour augmenter la puissance et la vitesse de calcul de son cerveau, comme il a jadis fabriqué le levier pour augmenter sa puissance musculaire.

Qu'est-ce qu'un ordinateur?

Le dictionnaire dit qu'un ordinateur est un calculateur. Or, un abaque, une additionneuse, vous et moi sommes des calculateurs. Ceci est vrai, bien entendu, mais la machine possède sur l'homme deux avantages essentiels :

- elle est plus rapide,
- elle a une plus grosse capacité de calcul.

En d'autres termes, un ordinateur n'oublie jamais, et la même instruction produit toujours les mêmes résultats. De plus, l'ordina-

teur ne suit pas le rythme de vie humaine, ce qui revient à dire qu'il ne s'arrête pas dans son travail en laissant des petits mots du genre « retour dans 5 minutes » ou bien « parti déjeuner »

Comment fonctionne un ordinateur?

Un ordinateur fonctionne très simplement, car c'est une machine on ne peut plus bête... rapide, mais bête! C'est une machine conçue pour effectuer rapidement des opérations arithmétiques. L'électronique interne ne semble pas horriblement complexe... elle l'est! Les équipements de téléinformatique sont bien effrayants également, mais la machine ne sait faire qu'une seule chose : additionner! Si on lui dit d'additionner, de soustraire, de multiplier, de diviser, ou de prendre une décision, tout ce qu'elle sait faire c'est additionner! De plus, elle est bête au point que « 1 » et « 1 » ne font pas 2, mais ceci est une autre histoire... d'arithmétique binaire. Bien sûr, elle possède certains « trucs » pour arriver à ses fins, mais pour celui qui l'observe travailler, il semble qu'elle passe au travers d'un processus extrêmement laborieux avant de pouvoir fournir une réponse. Seulement elle est rapide, et là réside toute sa puissance. On imagine mal qu'elle puisse effectuer une multiplication de 9 chiffres par 9, en moins de temps qu'il n'en faut pour prendre le crayon pour faire le calcul à la main... et pourtant!

La machine ne fait que ce qu'on lui dit de faire, et ne peut utiliser que les informations qu'on lui a transmises. Mais à la différence de l'idiot du village, elle a une mémoire prodigieuse et elle raisonne *logiquement* à très grande vitesse. Ainsi, si l'on veut faire fonctionner de manière satisfaisante un ordinateur, on doit tenir compte des points importants suivants :

- tout problème, travail ou information que l'on donne à la machine (entrée) doit être mis sous la forme la plus simple et la plus précise qui soit,
- tout problème, travail ou information que l'on demande à la machine ne doit s'appuyer que sur les données que possède la machine, ou que l'on va lui communiquer,
- les entrées doivent être faites dans un langage que la machine comprend,
- on doit lui préciser toutes les étapes qu'elle doit exécuter afin d'obtenir le résultat souhaité, un peu à la façon dont on enseigne à des élèves en se basant sur un programme défini.

Exactitude et précision des informations d'entrée

Maintenant que l'on vient d'examiner ce qu'un ordinateur peut fournir comme résultats à condition d'être correctement alimenté

en informations, on peut deviner aisément ce qui se passerait si on ne lui fournissait pas d'informations précises et exactes. Les anglais utilisent l'expression « GIGO (garbage in-garbage out) » pour désigner cette situation, que l'on pourrait traduire par une expression du style à question « idiot », réponse « idiot » ou « à entrée idiote, sortie idiote » dans ce cas.

Si, par exemple, un magasinier indique à l'ordinateur qui tient la comptabilité du magasin, qu'il a sorti 10 pièces, alors qu'en réalité il en a sorti 100, il se peut que la machine accepte son information si son indication du nombre de pièces en stock était supérieure à 100. Si au contraire, ce magasinier indique qu'il a sorti 100 pièces alors que la machine n'en avait que 10 en stock, cette dernière refuse l'information car elle sait qu'on ne peut soustraire 100 pièces d'un lot de 10.

Cet exemple montre que l'on doit être exact et précis, car si l'erreur reste logique, la machine peut l'accepter et le résultat peut être amusant, embarrassant ou catastrophique, suivant le sens de l'humour que chacun possède. On peut imaginer facilement qu'elle serait sa réaction si à la place du lave-vaisselle qu'elle a commandé, une personne recevait une presse hydraulique de 20 tonnes, à cause d'une erreur du service d'expédition sur le numéro de code.

Le manque d'exactitude peut également coûter cher. Si l'on suppose qu'un ordinateur coûte une somme X à l'achat, et que son entretien annuel coûte une somme Y , et si l'on suppose également que le traitement complet de toutes les transactions journalières prend 20 heures, cela laisse 4 heures de disponible. Seulement, si l'on suppose toujours que 50 % des entrées sont rejetées pour diverses raisons, il n'est pas besoin d'être ordinateur pour se rendre compte que des journées de 30 heures ne peuvent exister. Une solution consisterait à acheter un deuxième ordinateur de coût Z qui ne fonctionnerait qu'à 50 %, etc. Il est bien évident que l'on n'a pas les moyens de cette politique d'inexactitude.

Éléments d'un ordinateur

Un ordinateur numérique réalise une suite d'opérations arithmétiques sur des données, en suivant un programme d'instructions qui définit chaque étape.

Un ordinateur est calqué sur le comportement humain lorsqu'il traite un problème. L'information d'origine, enregistré d'une façon ou d'une autre, est fournie à la machine; c'est ce que l'on appelle une *information d'entrée*. La réponse finale est fournie par la machine; c'est ce que l'on appelle une *information de sortie*. Entre les deux se situe la zone de travail que l'on appelle l'unité centrale de traitement, dans laquelle les données et le pro-

gramme sont enregistrés et également dans laquelle s'effectuent les opérations de calcul, le tout supervisé par l'unité de commande.

La capacité de la mémoire interne de l'unité centrale de traitement n'est pas toujours suffisante pour conserver la totalité des programmes disponibles ainsi que les données. On rajoute donc des mémoires auxiliaires auxquelles l'unité centrale de traitement peut accéder rapidement.

Un ordinateur complet se compose donc d'une unité centrale de traitement entourée d'un certain nombre d'équipements appelés « périphériques », tel que le montre la figure 1.

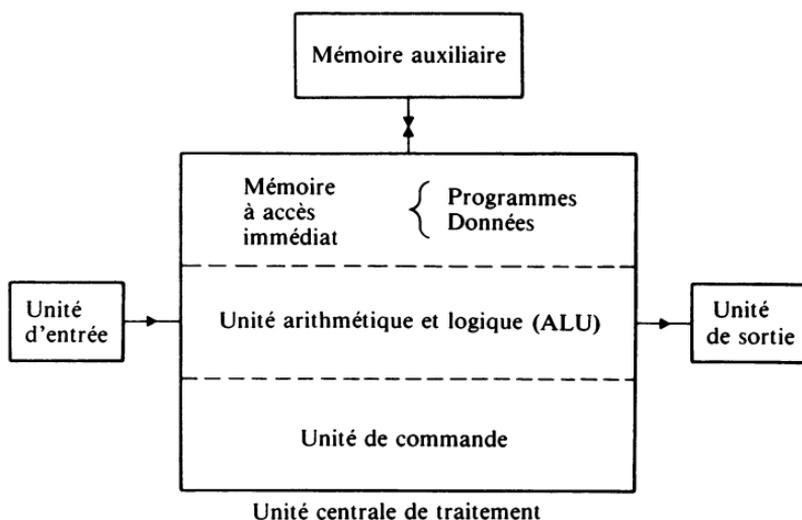


Fig. 1 Eléments d'un ordinateur.

Pour faire fonctionner un ordinateur, il est nécessaire d'avoir à sa disposition à la fois un ordinateur et ses équipements périphériques; c'est ce que l'on appelle le *matériel* de l'installation (hardware). Ce matériel est associé à des programmes; c'est ce que l'on appelle le *logiciel* (software) de l'installation.

L'unité centrale de traitement

L'unité centrale de traitement est le cœur de l'ordinateur, et doit pouvoir traiter des fonctions logiques et arithmétiques, sous le contrôle de l'unité de commande. Il est évident qu'elle doit posséder des circuits d'entrée et de sortie, ainsi qu'une mémoire pour y stocker les instructions, même si chaque partie possède sa mémoire temporaire spécifique, que l'on désigne généralement par « registre ».

La mémoire sert à enregistrer des instructions et comme elles doivent pouvoir être lues indéfiniment, il ne faut pas que la lecture de la mémoire détruise l'information qu'elle contient. Si l'on utilise une mémoire à semi-conducteurs, celle-ci peut prendre la forme d'une mémoire morte (ROM). Ou bien, si l'on a besoin des données que de façon temporaire, lors d'un calcul par exemple, il n'est plus nécessaire de les conserver une fois le calcul terminé. Dans ce cas, si l'on utilise une mémoire à semi-conducteurs, celle-ci peut prendre la forme d'une mémoire à accès direct (RAM).

Les instructions

L'organisation d'un ordinateur dépend dans une large mesure du format des instructions du programme. Un exemple, à adresse unique, est donné à la figure 2.

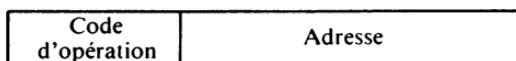


Fig. 2 Exemple de format d'adresse.

Le *code d'opération* spécifie l'opération ou la suite d'opérations à effectuer sur les données que contient la mémoire, dont l'adresse est spécifiée par un nombre figurant dans la partie adresse de l'instruction.

Dans un format incluant deux adresses, le code d'opération va tenir compte des deux adresses, ce qui permet, par exemple, de réaliser l'instruction « additionner le contenu d'une adresse à celui de l'autre adresse ». Un format à trois adresses permet d'ajouter l'adresse du résultat dans cette mémoire.

Les formats décrits précédemment peuvent également contenir des indications concernant un registre, dont le contenu peut servir à modifier la partie adresse de l'instruction. Ceci est conçu de façon à pouvoir utiliser le même jeu d'instructions pour travailler sur des données localisées à des adresses différentes que celles figurant sur les instructions. L'addition du contenu du registre à la partie adresse de l'instruction, fournit la nouvelle adresse permettant d'obtenir les données sur lesquelles on veut travailler.

Les mémoires

Les différentes utilisations des mémoires sont les suivantes :

- enregistrement du programme,
- enregistrement des données,

- enregistrement des paramètres nécessaires,
- enregistrement des résultats partiels de calcul,
- enregistrement des résultats finals avant l'impression.

Les deux facteurs les plus importants d'une mémoire sont sa rapidité et sa capacité. Or ces deux facteurs sont opposés, car une mémoire rapide est à faible capacité et une mémoire à forte capacité est lente. Un compromis est donc à trouver pour un coût donné. On peut utiliser une mémoire rapide à faible capacité, une RAM (mémoire à accès direct), par exemple, pour mémoriser les données immédiates de l'opération en cours, et utiliser une mémoire lente à forte capacité, comme une bande ou un disque magnétique, pour mémoriser le reste des données. Les données sont transférées d'une mémoire à une autre suivant les instructions du programme.

Unité arithmétique et logique (ALU)

L'unité arithmétique et logique comprend principalement un certain nombre de registres raccordés entre eux par des circuits logiques, pour réaliser les opérations que lui dicte l'unité de commande. Un registre est une mémoire qui contient un ou plusieurs nombres nécessaires aux opérations à effectuer.

Le registre le plus important est l'accumulateur, dont le rôle peut être illustré par l'exemple qui suit : une instruction de *remise à zéro et d'addition* remet à zéro le contenu de l'accumulateur et y insère le nombre provenant de l'adresse spécifiée par l'instruction; une instruction *d'addition* ajoute un nombre à celui figurant dans l'accumulateur, de telle sorte que ce dernier contient maintenant le résultat de l'addition des deux nombres. Ceci correspond au principe de fonctionnement de l'ALU (unité arithmétique et logique). De plus, certains registres supplémentaires peuvent y être incorporés, suivant la spécificité des opérations à effectuer.

L'unité de commande

Les parties essentielles de l'unité de commande sont les suivantes :

- le *registre d'instructions* qui contient les instructions d'exécution les plus courantes,
- le *décodeur* qui transforme le code d'opération de l'instruction en signaux logiques compréhensibles pour les circuits logiques,
- le *registre d'adresses* qui dirige la partie adresse de l'instruction vers la mémoire correspondante,
- le *compteur ordinal* qui contient l'adresse de l'instruction en cours. Lorsque l'instruction en cours est exécutée, le compteur

ordinal est incrémenté, c'est-à-dire qu'on lui rajoute un « 1 ». Le contenu correspond alors à l'adresse de l'instruction suivante, que le registre d'adresses va chercher dans la mémoire. Cette méthode permet de s'assurer que les instructions sont exécutées dans l'ordre.

Les unités périphériques d'entrée

Les unités d'entrée reçoivent le programme par l'intermédiaire d'un support approprié, lecteur de bande ou de carte perforée, ou encore « télétype » (*). Les données sont également transmises par le même support, pour être stockées dans une mémoire en attendant que l'opération soit exécutée. Il n'y a aucune restriction d'utilisation d'unités d'entrée de différents types, car l'unité d'entrée sert de mémoire tampon entre la vitesse d'introduction des données dans l'ordinateur et la vitesse de fonctionnement de celui-ci. Les unités d'entrée les plus courantes sont les suivantes :

- *lecteur de bande perforée*

Une bande de papier d'une largeur de 25 mm environ, peut contenir jusqu'à huit trous dans le sens transversal. Ces trous correspondent à un code de représentation de chiffres et de lettres ainsi que de caractères spéciaux. Une cellule photo-électrique lit ce code, à raison de 1 000 caractères par seconde.

- *lecteur de carte perforée*

Une carte perforée de 8 × 20 cm contient 80 colonnes. Chaque colonne représente un caractère grâce à un ensemble de trous perforés. Une cellule photo-électrique peut lire ces cartes à raison de 800 cartes par minute, ce qui correspond approximativement à 1 000 caractères par seconde.

- *télétype*

On utilise généralement un télétype en tant que terminal distant. L'entrée des informations peut se faire au clavier ou à l'aide d'une bande déjà perforée que l'appareil peut lire. La vitesse typique de transmission est de 10 caractères par seconde.

- *console*

Une console de visualisation présente un certain nombre de limitations, et ne peut remplacer un télétype. Elle sert principalement à l'opérateur de commande de l'ordinateur pour entrer et visualiser ses données.

* Ce terme qui est en fait une marque désigne une imprimante perforatrice avec clavier similaire à une unité de télex.

Les unités périphériques de sortie

Les unités de sortie transmettent les données qui viennent d'être traitées, de la mémoire de l'ordinateur à un support de sortie approprié, tel qu'un perforateur de bande ou de carte, une imprimante ou un modem (modulateur, démodulateur) de transmission de données. Cette unité de sortie sert de mémoire tampon entre la vitesse de sortie des données de l'ordinateur, et la vitesse de transmission du support employé. Les unités de sorties les plus courantes sont les suivantes :

- *perforateur de bande ou de carte*

Un perforateur de bande ou de carte est utilisé à la fois comme unité de sortie, car il fournit une bande ou une carte perforée sur laquelle le résultat du travail de l'ordinateur est inscrit, et comme unité d'entrée, la bande ou la carte étant ensuite lue par le lecteur de bande ou de carte perforée d'entrée.

- *imprimante*

Une imprimante est capable d'imprimer un grand nombre de caractères à plus ou moins grande vitesse selon la nature du bloc imprimant. Les résultats peuvent s'imprimer ligne par ligne sur certains modèles rapides, avec une cadence de 1 000 lignes à la minute de 160 caractères chacune.

- *télétype*

Une télétype peut servir en remplacement d'une imprimante. La vitesse de transmission est bien plus faible dans ce cas-là.

- *console*

Une console présente un certain nombre de limitations en tant qu'unité de sortie. Elle sert principalement à transmettre un message à l'opérateur de commande de l'ordinateur.

Applications

Les ordinateurs s'appliquent à toutes les branches de l'activité humaine. Si certaines applications ne sont pas convaincantes, il en est d'autres où l'ordinateur a fait preuve de sa puissance. Ces applications se subdivisent en deux domaines principaux :

- le calcul scientifique, et
- la gestion d'entreprise.

Selon les domaines d'activité, l'ordinateur est conçu différemment. Dans un ordinateur scientifique, on favorise la partie « calcul » au profit de la partie « mémoire » alors que dans un ordinateur de gestion, on favorise la partie « mémoire » au profit de la partie « calcul »

Le développement des ordinateurs

Les ordinateurs subissent dans leur évolution un certain nombre d'étapes de développement, dont certaines présentent un intérêt particulier. L'introduction, il y a quelques années, du mini-ordinateur permit de réaliser une économie notable sur la conception des systèmes industriels de contrôle, car il s'adaptait mieux à cette application particulière. Plus récemment, le microprocesseur a été lancé sur le marché. Généralement, il est constitué d'un seul circuit intégré en technologie MOS, que l'on complète par quelques circuits intégrés de mémoire et d'interface d'entrée-sortie. *L'architecture de principe de ces diverses formes d'ordinateurs subsiste : taille et vitesse changent.*

le tour des mémoires

La mémoire à accès immédiat d'une unité centrale de traitement est réalisée à partir d'une matrice d'éléments de mémoire à accès rapide, dans laquelle il est possible d'accéder à un élément de mémoire en l'adressant par ses coordonnées X et Y de la matrice.

Dans les gros ordinateurs, cette matrice était constituée de tores de ferrite. Toutefois, la progression des mini-ordinateurs et des micro-ordinateurs, de pair avec l'avance technologique des semi-conducteurs, ont conduit l'utilisateur à se servir concurremment des mémoires à semi-conducteurs et des mémoires à tores magnétiques.

La mémoire à accès immédiat est raccordée à l'unité centrale de traitement à travers une mémoire tampon, laquelle peut être utilisée pour réécrire les données dans la mémoire, en cas de destruction après une opération de lecture.

Les mémoires à semi-conducteurs comprennent les RAM (mémoires à accès direct), les ROM (mémoires mortes), les PROM (mémoires mortes programmables), les EPROM (mémoires mortes programmables et effaçables) et les EAROM (mémoires mortes altérables électriquement). Elles font appel à la technologie bipolaire, MOS, CMOS, (MOS complémentaire), NMOS (MOS à canal N), I²L (logique à injection), aux bulles magnétiques, aux CCD (dispositifs à couplage de charge) et plus récemment encore aux mémoires holographiques.

La capacité de mémoire peut être augmentée en utilisant des mémoires de masse à temps d'accès lent, comme les bandes, les disques ou les tambours magnétiques. Ces mémoires sont généralement définies comme des mémoires auxiliaires et elles sont considérées comme des éléments périphériques d'un ordinateur. Ce chapitre passe en revue les principes et les technologies des différents types de mémoires actuellement disponibles.

Les mémoires à tores de ferrite

Une mémoire à tores de ferrite est composée d'une matrice de tores de ferrite soutenus par des fils. Les tores sont assemblés par groupes, chaque groupe correspondant à un mot (ou byte) dont la

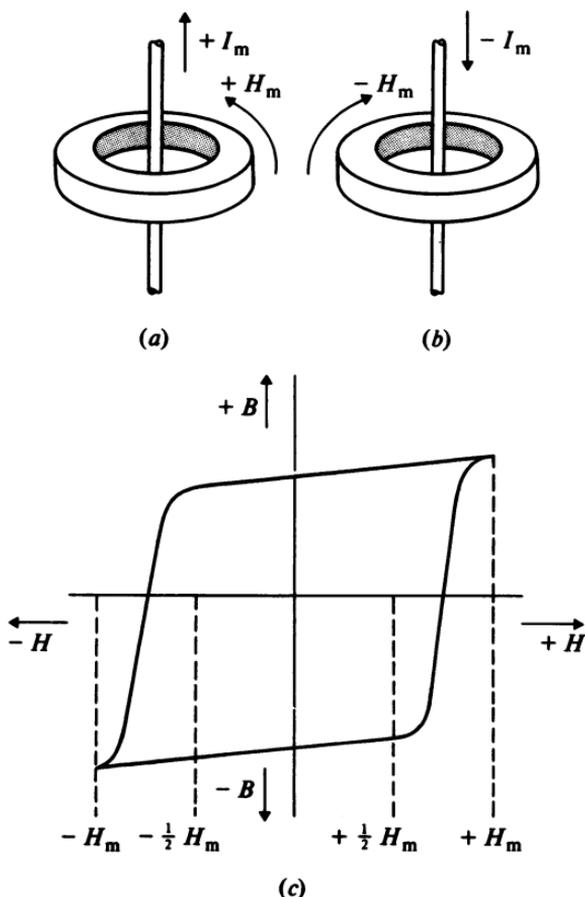


Fig. 3 Magnétisation d'un tore de ferrite :

- (a) magnétisation positive,
- (b) magnétisation négative,
- (c) caractéristique de magnétisation.

longueur permet de mémoriser des données ou les instructions d'un programme. La longueur du mot dépend de l'ordinateur, mais généralement elle est comprise entre 8 et 48 éléments binaires. La capacité totale de la mémoire est généralement définie en k mots ou en k bits, un k mot correspondant à 1 024 mots (2^{10}) (le k bits correspond à 1 024 bits). L'écriture la plus fréquente est : mémoire de xK . Le diamètre d'un tore est approximativement de 0,4 mm.

Le tore de ferrite peut être magnétisé dans un sens ou un autre — car il a deux états stables de magnétisation — en envoyant un courant de magnétisation I_m dans un fil traversant le tore. Suivant le sens du courant, le tore est magnétisé positivement ou négativement, tel que l'indique la figure 3.

Chaque tore de la matrice est traversé par deux fils X et Y , ce qui permet de sélectionner un tore particulier en envoyant sur chaque fil la moitié du courant de magnétisation (fig. 4).

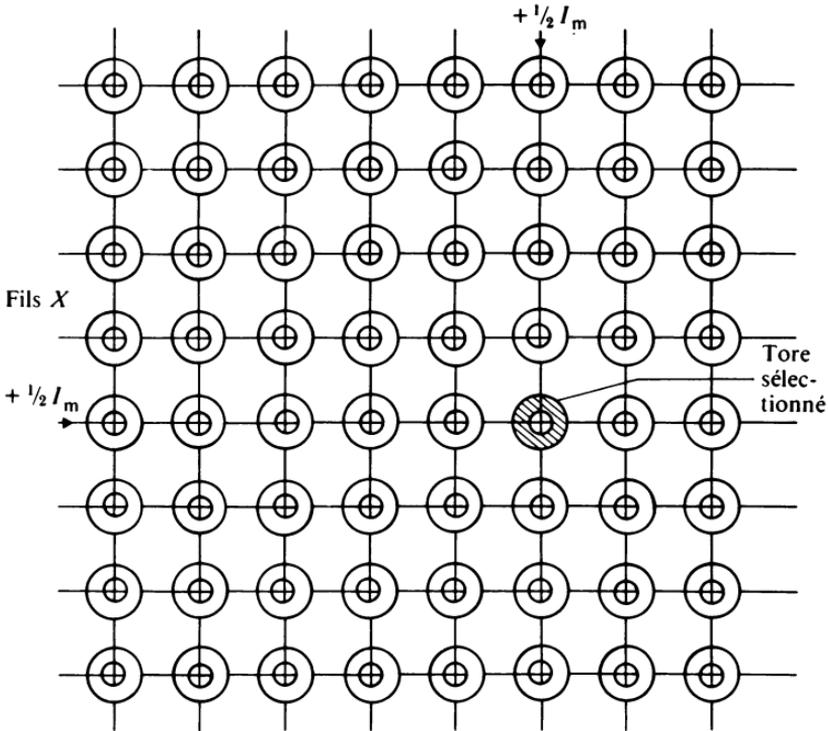


Fig. 4 Sélection d'un tore dans la matrice.

Les matrices de tores sont généralement disposées en plans parallèles, de telle sorte que le même tore de chaque plan constitue un mot binaire (fig. 5). Un plan additionnel sert de contrôle de parité à un mot de 8 éléments binaires.

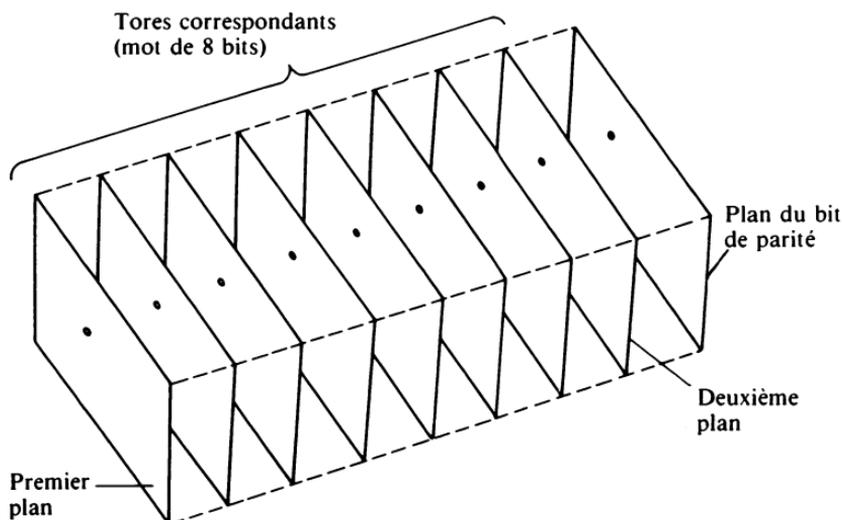


Fig. 5 Représentation d'un mot de 8 éléments binaires.

Chaque tore est traversé par quatre fils, identifiés comme suit : fil *X*, fil *Y*, fil *Z* d'inhibition, fil *S* de lecture.

Dans chaque matrice, il n'y a qu'un fil *Z*, ainsi qu'un seul fil de lecture, traversant tous les tores de la matrice, comme l'indique la figure 6.

Lecture et écriture d'une mémoire

Lorsqu'on envoie simultanément sur les fils *X* et *Y*, un courant égal à la moitié de la valeur de magnétisation d'un tore, on magnétise le tore situé au point d'intersection des fils *X* et *Y*. Il est alors possible d'effectuer quatre opérations sur un tore magnétisé :

- *l'opération de lecture*

L'état du tore est lu en y écrivant un état logique 0 et en détectant un changement éventuel de son état. Le fil *S* de lecture passant à travers tous les tores d'une matrice sert à détecter le changement. Si l'état du tore change, un courant s'induit dans le fil *S*, c'est-à-dire que si le tore était initialement à « 0 », il n'y a aucun courant, et si le tore était initialement à « 1 », il y a induction d'un courant.

- *l'opération de régénération*

Après l'opération de lecture, tous les tores sélectionnés sont à « 0 », de telle sorte que si l'information qu'ils possédaient doit être conservée, il est nécessaire d'effectuer une régénération de cette information. Les fils *X* et *Y* sont alors traversés par un courant inverse de magnétisation destiné à magnétiser à l'état « 1 » les tores sélectionnés. Toutefois, certains de ces tores étant initia-

lement à « 0 », pour éviter que ces tores soient mis à « 1 », on envoie simultanément sur le fil Z d'inhibition un courant destiné à inhiber la magnétisation produit par les fils X et Y.

• *l'opération d'écriture*

Cette opération est identique à l'opération d'inhibition et l'on se sert du fil Z d'inhibition lorsque l'on désire écrire un « 0 ». L'opération d'écriture est toujours précédée d'une lecture.

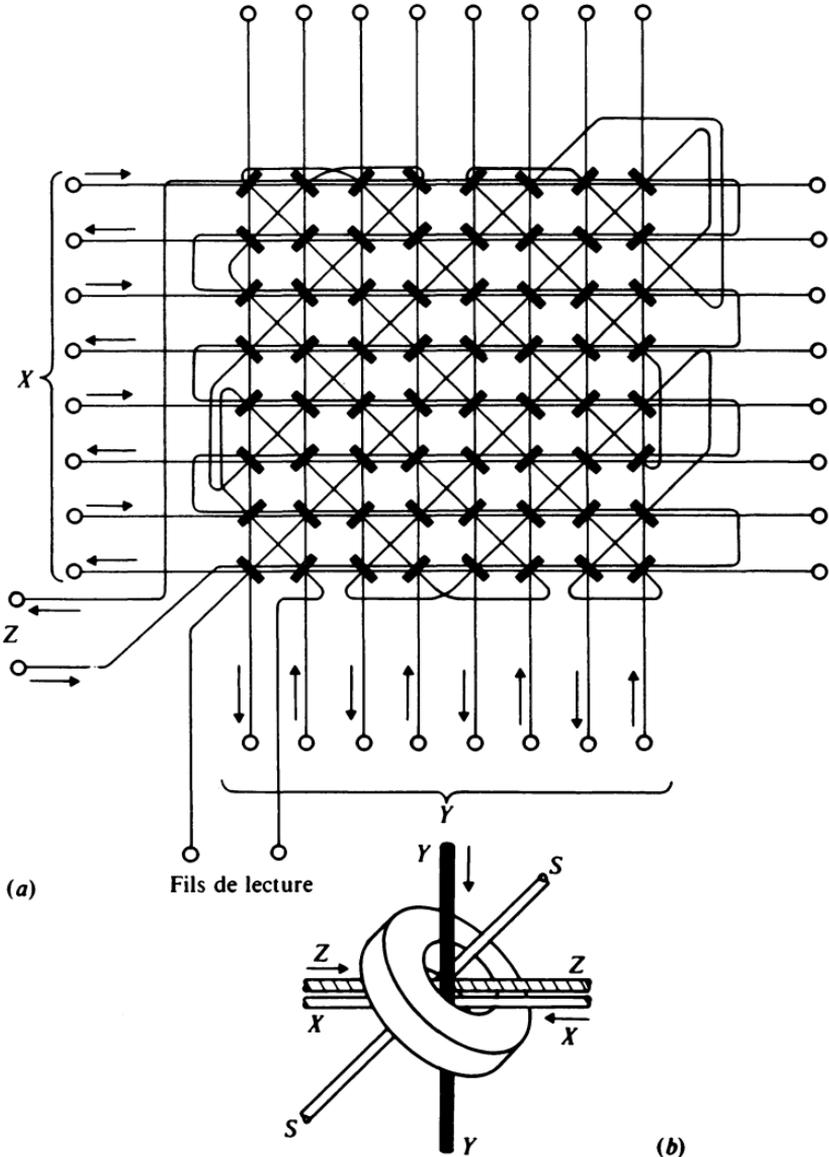


Fig. 6 Matrice de tores de ferrite avec son réseau de fils :
 (a) câblage d'une matrice,
 (b) câblage d'un tore.

- *la vérification de parité*

Cette méthode permet de vérifier le bon fonctionnement d'un transfert de données. Un bit de parité indique la parité du mot binaire, c'est-à-dire le nombre de « 1 » du mot. On vérifie ce bit après chaque transfert.

Une mémoire à tores est donc une mémoire à accès direct, car chaque emplacement de mémoire a une adresse particulière. Le temps d'accès est meilleur que $2 \mu\text{s}$ et il est le même quel que soit le tore considéré. Ce type de mémoire est très fiable, consomme peu et occupe peu de place. Il est cependant coûteux.

Mémoire à film mince

Un mince film d'alliage fer-nickel est déposé sur du verre, en présence d'un champ magnétique puissant. Le film mince agit alors comme un aimant qu'il est possible de magnétiser dans un sens ou un autre, pour représenter un « 0 » ou un « 1 » binaire. La construction d'une telle mémoire est identique à celle d'une mémoire à tores, à l'exception de bandes conductrices déposées sur du verre, qui servent de fils.

Il existe une ligne par bit, pour chaque rangée d'éléments d'un plan et un seul fil de lecture par plan. Une ligne par mot est utilisée pour sélectionner les éléments de chaque plan. Les éléments de mémoire sont placés en sandwich entre les bandes conductrices.

Quoique ce type de mémoire consomme moins qu'une mémoire à tores et que son temps d'accès est plus rapide ($0,2 \mu\text{s}$), le signal de sortie doit être considérablement amplifié pour être utilisable. Ce type de mémoire est peu ou pas utilisé.

Mémoire à semi-conducteurs

Il est possible de se procurer couramment un grand nombre de mémoires à semi-conducteurs, et quoique aucune standardisation n'ait eu lieu, il se dégage certaines tendances en matière de boîtiers et de brochage chez plusieurs constructeurs. La mémoire à semi-conducteurs idéale serait celle qui tout en occupant la plus petite place, aurait le temps d'accès le plus rapide et la consommation la plus faible qu'il soit. De plus, il faudrait qu'elle soit non volatile, c'est-à-dire capable de conserver ses informations, même en cas de coupure d'alimentation. Il n'y a toutefois que les ROM (mémoires mortes) qui réalisent cette condition.

Dans la conception d'une mémoire, il faut tenir compte du procédé technologique employé et de la conception de l'élément de mémoire. La technologie bipolaire permet d'obtenir des temps

d'accès rapides avec une faible densité d'intégration, alors que les technologies MOS et CMOS permettent un temps d'accès moins grand avec une forte densité d'intégration. Les mémoires diffèrent des circuits logiques, en ce sens que le dessin d'une mémoire consiste en une mosaïque de cellules élémentaires, ce qui permet d'obtenir une plus grande densité d'intégration, qu'avec les circuits logiques. Le temps de fonctionnement moins rapide des transistors MOS provient principalement de la capacité parasite constituée entre le transistor et le substrat, problème que les fabricants arrivent petit à petit à résoudre grâce à de nouvelles technologies, comme le silicium sur saphir (SOS), les NMOS et plus récemment encore les VMOS (MOS en V).

Toutes les cellules de mémoire peuvent être soit *statiques*, si l'information y demeure tant que l'alimentation est branchée, ou *dynamiques* si l'information, correspondant à une charge de condensateur, doit être périodiquement rafraîchie pour compenser les fuites de charge des condensateurs.

Mémoire RAM statique

La cellule élémentaire d'une telle mémoire est une bascule bistable à transistors bipolaires. Les cellules sont disposées en matrice, et chaque cellule peut être adressée par deux fils X et Y . Un mot peut être constitué par exemple, d'une rangée de plusieurs cellules de cette sorte, qu'il est possible d'adresser simultanément.

La figure 7a donne le schéma d'une cellule élémentaire de mémoire statique bipolaire. Lorsque les fils de sélection X et Y sont au niveau bas, le courant traverse le transistor $TR1$ ou $TR2$, suivant l'information contenue dans la bascule. Pour effectuer la lecture d'une cellule, on place les deux fils X et Y au niveau haut de façon à obliger le courant traversant l'un des deux transistors, à emprunter la ligne des données correspondante, fournissant ainsi l'indication de l'état de la cellule. Pour effectuer l'écriture d'une cellule, on place les deux fils X et Y au niveau haut, et on envoie sur la ligne de donnée le potentiel correspondant au niveau logique que l'on veut écrire.

Une cellule similaire à transistors MOS complémentaires est donnée à la figure 7b. Quoique cette cellule utilise des MOS complémentaires, ce sont les MOS à canal N qui sont employés pour l'adressage, autorisant ainsi des temps d'accès de 200 ns avec une consommation bien plus faible qu'avec la technologie à transistors bipolaires. Les lignes de données ne sont raccordées à la cellule de mémoire que lorsque les deux fils de sélection X et Y sont simultanément placés à un potentiel négatif.

Il est courant maintenant de trouver des mémoires RAM statiques de 4 k bits (4 096 bits), en boîtiers DIL à 16 ou 22 broches. Dans les années à venir, cette capacité sera certainement augmentée à des valeurs de 16 K, 32 K, etc.

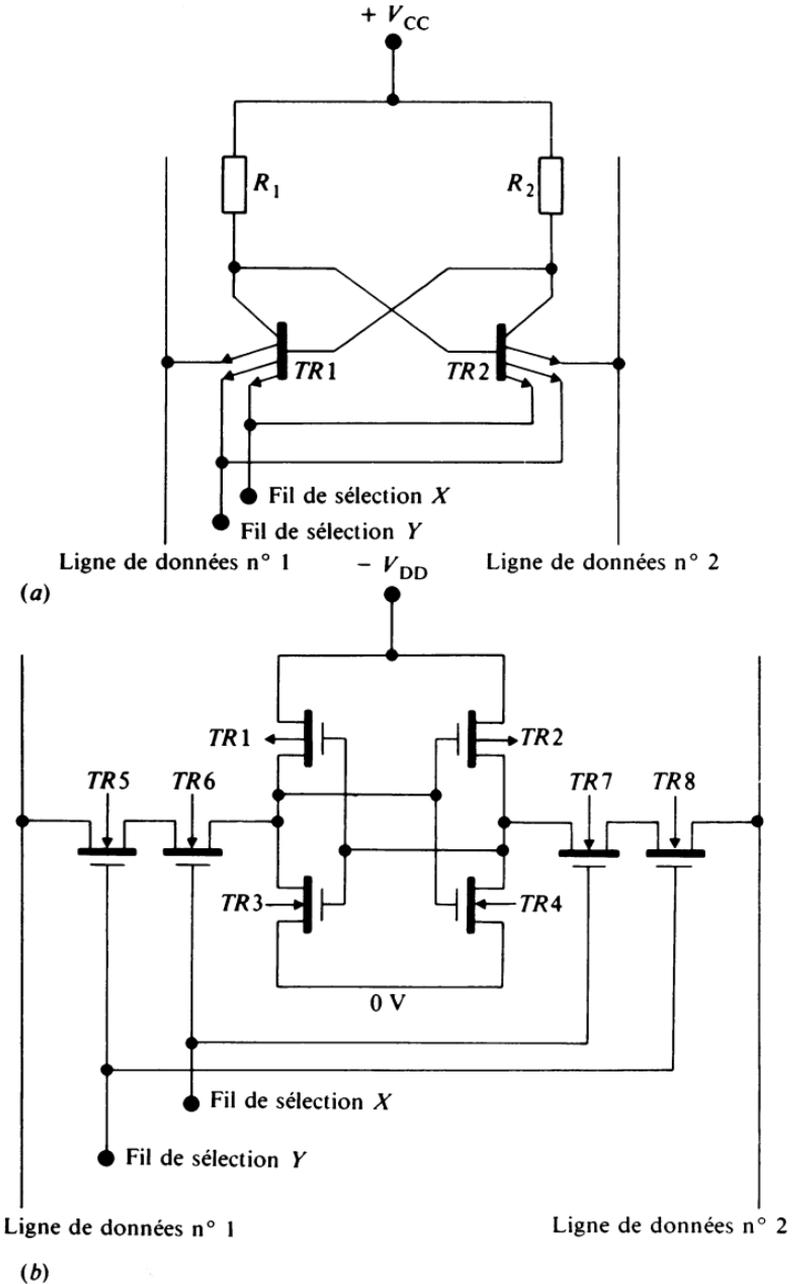


Fig. 7 Mémoire RAM statique :
 (a) à transistors bipolaires,
 (b) à transistors MOS complémentaires.

condensateur. Lorsque le transistor $TR2$ est passant, l'information aux bornes de C_1 peut être rafraîchie ou modifiée. La lecture de la cellule s'effectue en raccordant la ligne de lecture au potentiel négatif (niveau logique « 1 », ce qui entraîne la conduction de $TR3$). Si l'information emmagasinée par le condensateur C_1 est un « 1 » logique, le transistor $TR1$ devient passant à son tour, ce qui décharge la ligne de lecture. Si l'information emmagasinée par le condensateur C_1 est un 0 logique, le transistor $TR1$ reste bloqué et la ligne d'écriture n'est pas déchargée. A la fin du cycle de lecture, la valeur de l'information sur le fil de lecture est le complément de l'information emmagasinée par le condensateur. On rafraîchit la valeur de cette information après chaque cycle de lecture, en envoyant sur la ligne d'écriture le complément de l'information de la ligne de lecture.

Les mémoires dynamiques sont principalement constituées de transistors MOS, bien que l'on assiste à certains développements de technologies bipolaire du type Schottky ou I^2L . La figure 8 *b* donne le schéma d'une cellule dynamique composée de deux transistors bipolaires et de deux condensateurs. A l'état de repos, les diodes sont polarisées en inverse, ce qui isole la cellule des lignes de données. Le potentiel d'un condensateur est plus élevé que celui de l'autre, selon le transistor qui conduit. On adresse la cellule en plaçant la ligne de sélection de mot au niveau bas. La lecture de l'information peut se faire en détectant le courant circulant sur des lignes de données, et son écriture en appliquant les tensions adéquates aux lignes de données, de façon à forcer la cellule dans un état.

Les cellules de mémoire sont généralement disposées en matrice, et elles sont raccordées par l'intermédiaire des lignes de sélection X et Y aux décodeurs X et Y (fig. 9 *a*), dans laquelle chaque cellule est constituée par le montage des trois transistors MOS de la figure 8 *a*.

Des amplificateurs de régénération sont nécessaires pour rafraîchir les cellules toutes les 2 millisecondes. La figure 9 *b* donne le schéma d'un tel amplificateur. Les deux transistors $TR3$ et $TR4$ se comportent comme des résistances. L'information présente sur la ligne de lecture correspond au complément de celle contenue dans la cellule. Lorsque le transistor $TR1$ devient passant, il inverse à nouveau cette information et l'envoie sur la ligne d'écriture, de façon à la réécrire dans la cellule. Cette opération permet donc de régénérer l'information, colonne par colonne. L'écriture de nouvelles informations dans une cellule s'obtient en maintenant $TR1$ bloqué, et en envoyant directement l'information sur la ligne d'écriture.

La capacité totale d'une mémoire à semi-conducteurs peut être augmentée, en rajoutant des circuits additionnels de mémoire.

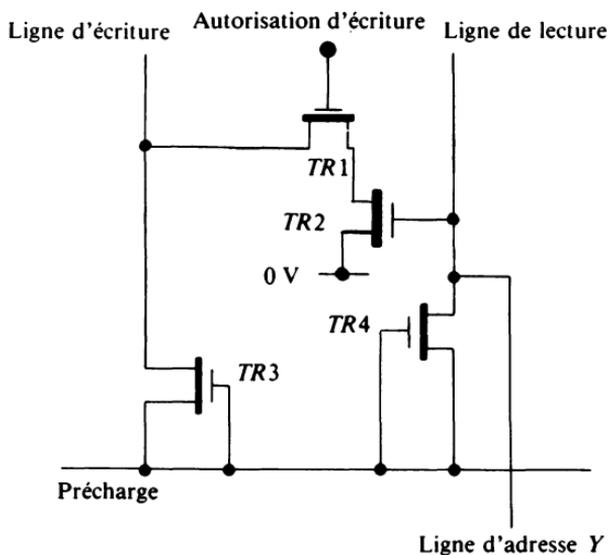
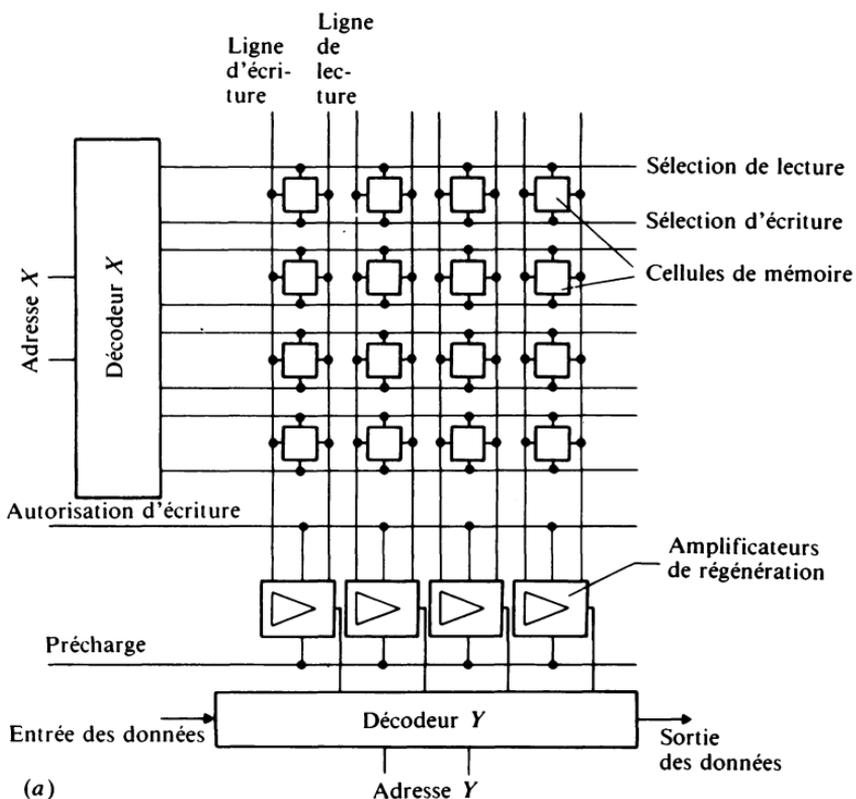


Fig. 9 Mémoire RAM dynamique :

(a) disposition et interconnexion,
 (b) amplificateur de régénération.

Mémoires ROM, PROM, EPROM et EAROM

Les mémoires mortes ROM sont construites de la même façon que les mémoires RAM à accès direct, de manière à pouvoir être adressées aussi rapidement. Toutefois, les mémoires ROM servent lorsque l'information que l'on désire y stocker n'est pas susceptible de changer, tel que des tables de valeurs, par exemple. Il existe diverses formes de mémoires ROM :

- *la mémoire morte* ROM programmée en usine à l'aide d'un masque, suivant des valeurs standard ou des spécifications à la demande;
- *la mémoire morte programmable* PROM, livrée non programmée au client, celui-ci pouvant la programmer suivant ces besoins. Cette formule est idéale pour le développement de nouveaux systèmes. La programmation s'effectue en faisant fondre un fusible de liaison à l'intérieur de chaque cellule de mémoire, de telle sorte qu'elle délivre le même niveau logique en permanence;
- *la mémoire morte programmable et effaçable* EPROM, dont le programme peut être entièrement effacé en exposant le circuit à une lumière ultraviolette intense. Généralement le dessus du circuit intégré est constitué par une plaque de verre, de façon à laisser passer cette lumière UV. Une fois la mémoire effacée, elle est à nouveau reprogrammable;
- *la mémoire morte altérable électriquement* EAROM, dans laquelle on peut effacer localement des données, sans toucher à l'ensemble de la mémoire, en envoyant une impulsion de tension élevée entre les broches de programmation.

Les mémoires mortes ROM sont essentiellement des mémoires à faible coût, à temps d'accès rapide et à grande capacité. Etant donné qu'elles doivent mémoriser un ensemble prédéterminé de signaux logiques, il n'est pas nécessaire que chaque cellule de mémoire soit constituée par une bascule. Un exemple simple de matrice à diodes est illustré à la figure 10 a, et l'on voit qu'elle suffit à réaliser une mémoire morte. Il suffit d'appliquer une tension positive à la ligne d'adresse appropriée, et les signaux logiques de sortie apparaissent sur les lignes de données : un « 0 » logique correspond à l'absence de diode et un « 1 » logique correspond à la présence d'une diode. Le fabricant change la disposition des diodes en changeant de masque de gravure. La figure 10 b donne un autre exemple de mémoire morte utilisant des transistors MOS. Dans ce cas, seuls les transistors MOS ayant leur grille raccordée à la ligne d'adresse sélectionnée, deviennent passants lorsque l'on applique un niveau logique « 1 » sur cette ligne, et délivrent alors un niveau logique « 0 » sur les lignes de données. Les autres transistors dont la grille n'est pas raccordée, délivrent en permanence un niveau logique « 1 » sur les lignes de données. Les mémoires

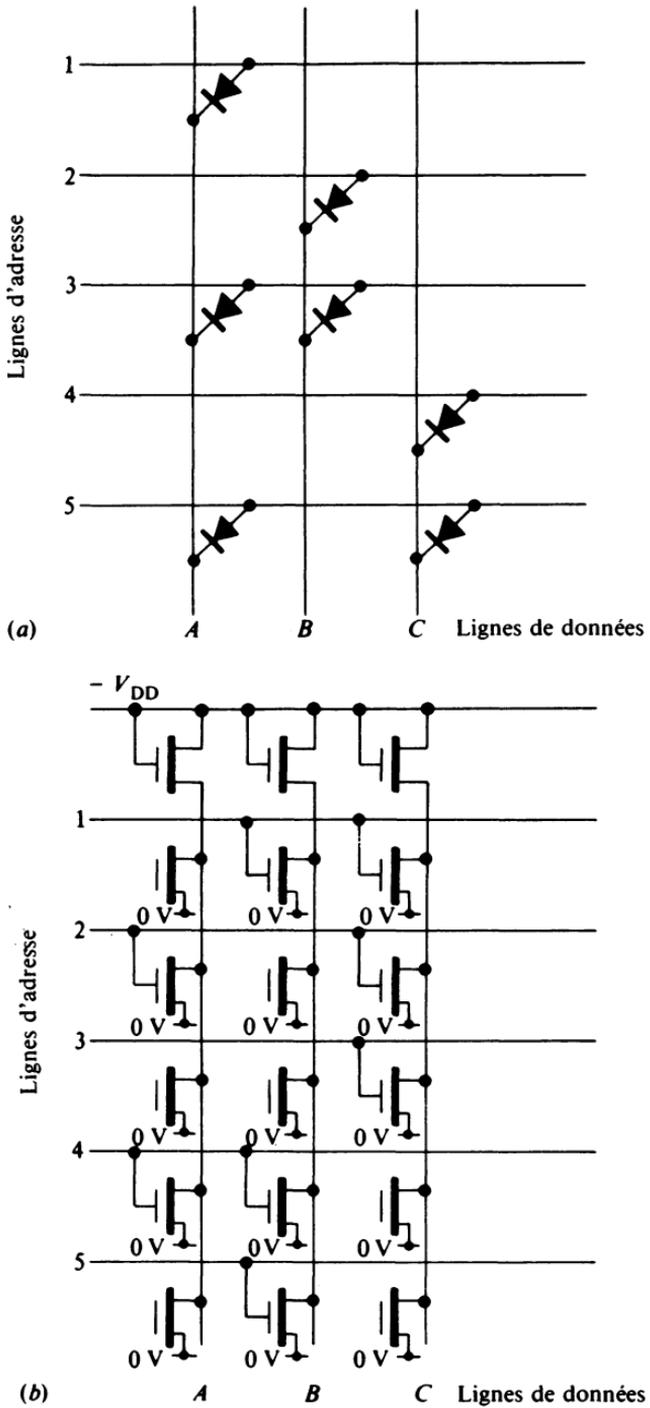


Fig. 10 Mémoires ROM à programmation par masque :

- (a) à matrice à diodes,
- (b) à matrice à transistors MOS.

mortes ROM sont couramment disponibles en versions de 16 K (16 k bits) et de 32 K (32 k bits).

Les mémoires mortes programmables PROM à fusible, sont constituées d'une matrice de diodes ou de transistors, où chaque diode ou transistor est alimenté en série avec un fusible. On programme une telle mémoire en faisant fondre les fusibles en série avec les diodes qui ne sont pas nécessaires dans la matrice, en envoyant une impulsion de courant. La figure 11 *a* montre une matrice à diodes et à fusibles de liaison. Une fois le programme enregistré dans la mémoire, il n'est plus possible de le changer, sauf si l'on désire supprimer quelques diodes supplémentaires. Une autre solution consiste à disposer d'une matrice de diodes montées tête-

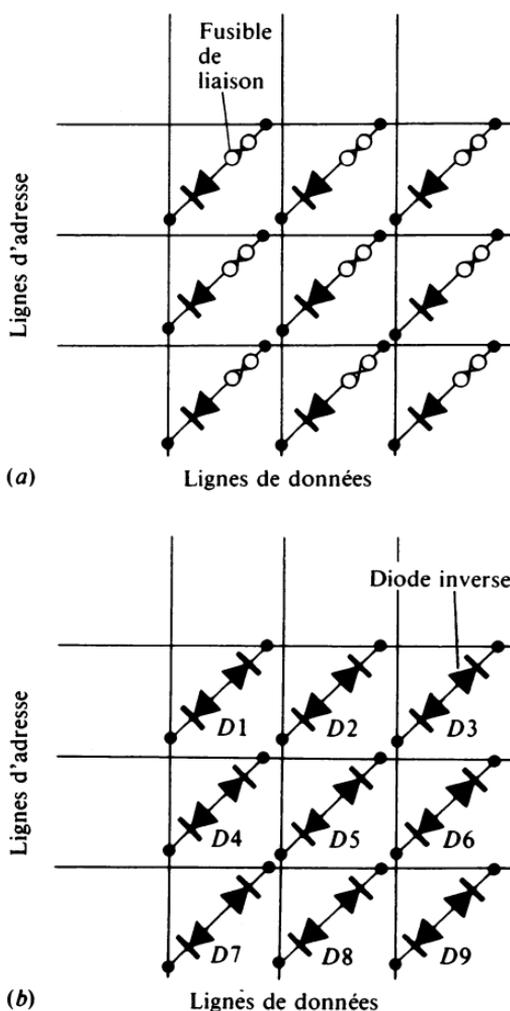


Fig. 11 Mémoires PROM :

- (a) à fusibles de liaison,
- (b) à diodes inverses.

bêche, ce qui, contrairement à la matrice à diodes et fusibles de liaison, entraîne une non-conduction des diodes avant programmation. Dans ce cas, on programme ce type de mémoire en appliquant une tension inverse suffisamment élevée aux diodes dont on désire obtenir le claquage.

Les mémoires mortes programmables et effaçables EPROM sont totalement effaçables, par exposition du circuit à une lumière ultraviolette pendant 10 minutes. Il est alors possible de les reprogrammer de la même manière que les mémoires programmables précédentes. Une mémoire EPROM fonctionne en piégeant une décharge dans la grille flottante d'un transistor MOS, tel que l'indique la figure 12. Les mémoires EPROM se trouvent couramment en version de 8 K (8 k bits).

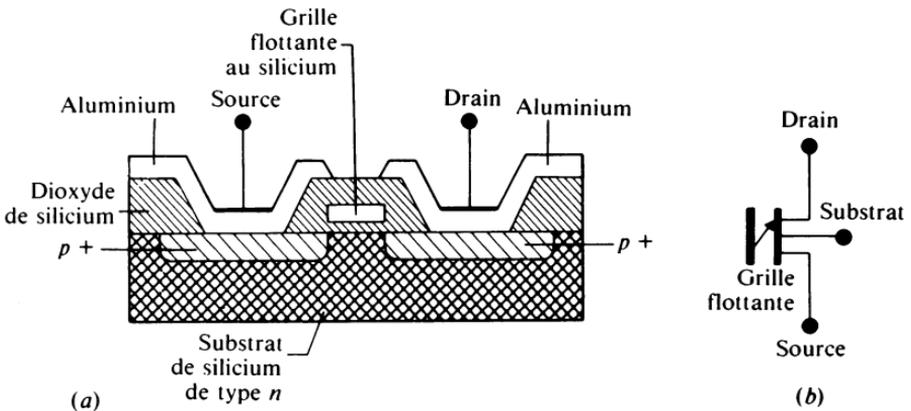


Fig. 12 Mémoire EPROM à transistors MOS à grille flottante :

- (a) coupe d'un transistor MOS à grille flottante,
- (b) symbole.

Les mémoires mortes altérables électriquement EAROM sont altérables localement en envoyant une impulsion de tension élevée entre les fils de programmation. De ce fait, on peut modifier le contenu d'un mot sans affecter le reste de la mémoire. Les mémoires EAROM sont couramment disponibles en versions de 32×16 bits, $2 \text{ k} \cdot 8$ bits et $1 \text{ k} \cdot 4$ bits.

Mémoires auxiliaires

Etant donné la capacité limitée des mémoires à accès immédiat, il est souvent nécessaire de disposer d'un supplément de mémoire, pour épauler la mémoire de l'unité centrale. Cette mémoire auxiliaire permet le stockage du volume important de données que peut traiter un ordinateur, mais elle présente un temps d'accès plus long. En règle générale, le prix de ce type de mémoire est proportionnel à son temps d'accès.

Les mémoires auxiliaires utilisent en général un support magnétique sur lequel les données sont enregistrées. Les informations sont écrites ou lues sur le film d'oxyde magnétique, en changeant l'orientation du champ magnétique produit par une tête magnétique de lecture et d'écriture. Le système d'enregistrement le plus répandu est le système d'enregistrement longitudinal, dans lequel le champ de magnétisation est parallèle au plan de l'oxyde magnétique. La figure 13 donne un exemple de tête magnétique de lecture et d'écriture. Lorsque le courant traverse l'enroulement d'écriture, la tête d'écriture se magnétise et le flux magnétique influence le film magnétique dans la région de l'entrefer de la tête d'écriture. Un courant de sens opposé entraîne une orientation opposée du flux magnétique, ce qui correspond aux deux états logiques « 0 » et « 1 ». La lecture du film magnétique se réalise en le déplaçant devant la tête de lecture, ce qui induit une force électromotrice dans l'enroulement de lecture.

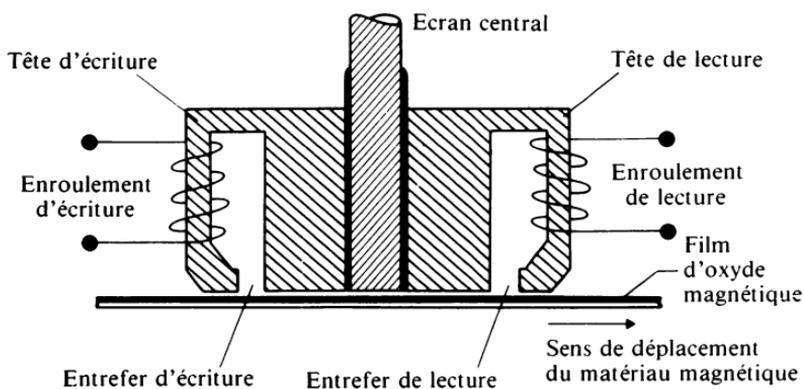


Fig. 13 Tête magnétique de lecture et d'écriture.

Les trois types de mémoires auxiliaires les plus répandues sont les mémoires à tambour magnétique, à bande magnétique et à disques magnétiques.

Mémoire à tambour magnétique

Un tambour magnétique, est un cylindre recouvert d'une pellicule d'oxyde ferromagnétique, tournant à vitesse constante autour d'un axe vertical. La surface de l'oxyde est divisée en plusieurs pistes horizontales ayant chacune leur tête de lecture et d'écriture, et en secteurs verticaux dont les limites sont fixées par la longueur des mots. De ce fait, il est possible de sélectionner un mot en indiquant au mécanisme d'entraînement, le numéro de la piste et du

secteur concerné. La disposition des données à la surface d'un tambour est illustrée à la figure 14 a. Si les données à transférer sont stockées sur plus d'une piste, un temps de commutation d'une piste à l'autre est à ajouter au temps de lecture. Ce temps, de commutation, qui peut valoir l'équivalent d'une révolution du tambour, peut être réduit en décalant le début du mot sur la piste suivante.

Il existe trois positions possibles des têtes de lecture et d'écriture :

- *têtes fixes*

Cette position comprend une tête par piste. Un entrefer constant est maintenu entre les têtes et le tambour magnétique dont la température est réglée avec précision, de façon à éviter le contact d'une tête avec le tambour en cas de dilatation de ce dernier.

- *têtes mobiles*

Cette disposition comprend une tête pour plusieurs pistes, positionnée par électro-aimant. Ce système présente un temps d'accès plus lent que le système à têtes fixes, mais il est bien meilleur marché.

- *têtes volantes*

Cette disposition comprend une tête par piste, pouvant se déplacer à la surface du tambour, sur coussin d'air. Un contact accidentel avec le tambour est minimisé, car le tambour monte sur son axe au fur et à mesure qu'il augmente de vitesse, ce qui permet de dépla-

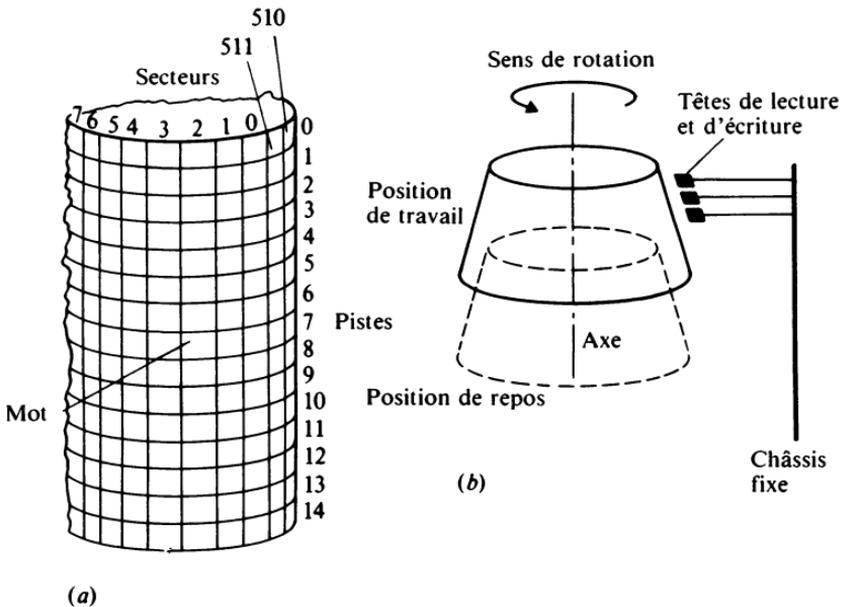


Fig. 14 Mémoire auxiliaire à tambour magnétique :

- (a) disposition à la surface du tambour,
- (b) système à têtes volantes.

cer les têtes quand il est en position de repos. La figure 14 *b* montre ce type de disposition qui permet d'obtenir un temps d'accès plus court, de pair avec une plus grande densité d'informations. Une telle mémoire est par contre plus onéreuse.

Mémoire à bande magnétique

La bande magnétique de ce type de mémoire est similaire à celle utilisée dans les magnétophones grand public, à la différence près qu'elle est plus large et de bien meilleure qualité. Elle est constituée par un support en vinyl recouvert d'une pellicule d'oxyde ferrique, enroulée en bobines de 50, 200, 500 et 1 000 mètres. La bande peut stocker l'équivalent de 1 000 caractères au centimètre et elle prend place dans un lecteur de bande, comme celui de la figure 15. Le mécanisme d'entraînement de la bande est constitué par un galet qui appuie la bande sur un cabestan en rotation permanente. L'arrêt de la bande s'obtient en écartant le galet de la bande et en freinant le moyeu de la bobine d'alimentation. Ce mécanisme doit être capable de démarrer et d'arrêter brutalement la bande, sans qu'elle subisse de dommage.

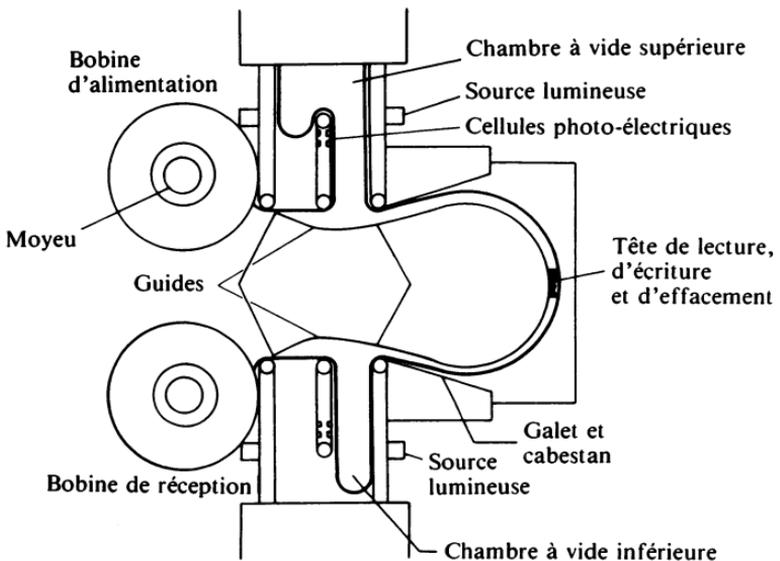


Fig. 15 Disposition d'un lecteur de bande magnétique.

Les informations sont stockées sous forme de trames (fig. 16). L'on voit qu'une trame d'une bande à sept pistes se compose de six bits de données et d'un bit de parité. Pendant l'opération de lecture, la parité de la trame est vérifiée avec le bit de parité, alors que pendant l'opération d'écriture, la parité de la trame est véri-

fiée par une lecture immédiate des éléments binaires écrits. Les données sont transférées par blocs entre l'unité centrale et la mémoire auxiliaire, avec des espaces de séparation interblocs, permettant à la bande d'acquérir la vitesse qui convient aux opérations de lecture et d'écriture.

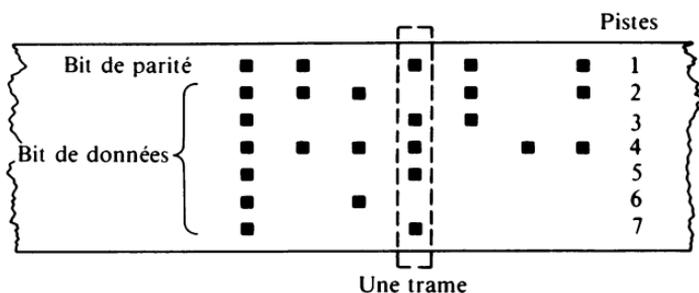


Fig. 16 Enregistrement des données sur une bande magnétique.

La disposition des têtes dans le lecteur de bande permet de réaliser les trois opérations : effacement, écriture et lecture. Pendant l'écriture, les données précédemment enregistrées sont effacées, et les nouvelles données sont écrites en série, trame par trame et bloc par bloc. Elles sont ensuite vérifiées grâce à la tête de lecture. Le temps d'accès d'une telle mémoire est plutôt long, — il peut être de plusieurs minutes, — mais son coût est relativement faible.

Mémoire à disques magnétiques

Une mémoire à disques magnétiques est constituée généralement d'une empilage de 6 disques, montés dans un coffret de transport comprenant le mécanisme d'entraînement, les têtes de lecture et d'écriture et l'électronique de commande. Un ensemble de six disques, tel que le montre la figure 17 a possède dix surfaces d'enregistrement magnétique en oxyde ferrique, les deux surfaces extérieures ne servant pas. Chaque surface active possède sa propre tête d'écriture et de lecture, montée sur un bras métallique. Lorsque les disques ont atteint leur vitesse de rotation normale, les têtes se déplacent et se positionnent individuellement sur leurs faces respectives par simple pivotement du bras qui les soutient. Chaque surface active comprend une centaine de pistes concentriques différentes. On désigne par le mot « cylindre », la même piste prise sur l'ensemble des dix surfaces actives (fig. 17 b). Chaque piste se divise en huit blocs ayant chacun la même capacité de mémoire. Les données s'écrivent et se lisent à l'intérieur d'un cylindre, et on y accède en adressant le cylindre et la tête correspondante.

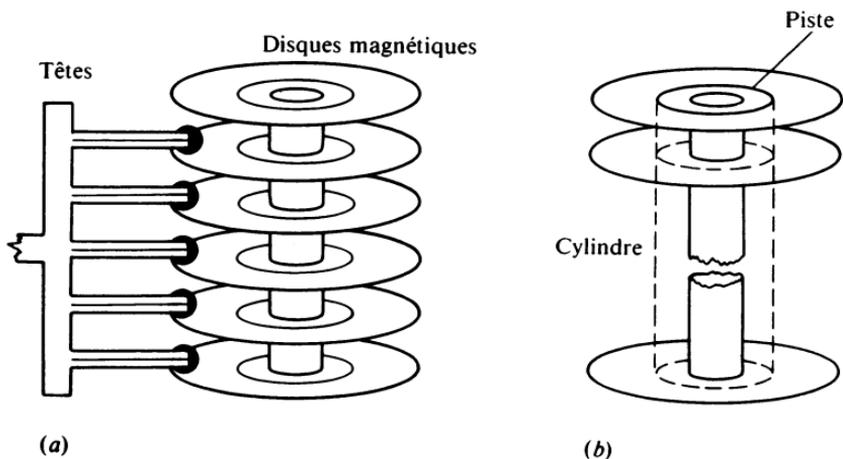


Fig. 17 Mémoire à disques magnétiques :

(a) disposition des têtes et des disques,

(b) constitution d'un cylindre

Un ensemble de six disques peut contenir jusqu'à cent millions de caractères. Le temps d'accès est rapide, puisqu'il est de 100 ms, mais le coût d'une telle mémoire est élevé. Récemment, les fabricants ont développé les disques souples pour répondre au même besoin de mémoire auxiliaire. Ces disques sont identiques du point de vue fonctionnement, mais ils sont plus petits et souples, comme leur nom l'indique. Leur coût est bien plus faible et ils s'adaptent très bien aux mini-ordinateurs modernes.

concevoir sa programmation

Tout problème devant être résolu par un ordinateur, doit pouvoir être écrit en une suite d'étapes bien précises, que l'on appelle un *algorithme*. L'algorithme qui permet à la machine de trouver la solution d'un problème, comprend un nombre fini d'instructions que celle-ci doit exécuter dans un ordre prédéterminé. Cette liste d'instructions peut se représenter sous la forme d'un diagramme composé de symboles associés les uns aux autres, que l'on appelle un *organigramme*.

Un organigramme de programmation correspond à une description détaillée du programme à utiliser pour résoudre un problème particulier, et il reflète invariablement la nature du langage utilisé par la machine. La réalisation d'un organigramme présente plusieurs avantages :

- obligation d'analyser le problème avant de tenter de produire une solution,
- présentation claire et simple de la façon dont le problème doit être résolu,
- simplification dans la recherche des erreurs de la solution, grâce à l'enregistrement qui en est fait,
- utilisation à des fins pédagogiques, pour décrire à d'autres personnes les opérations qui viennent d'être exécutées.

L'organigramme

Lorsque l'on écrit un organigramme, on doit nécessairement se poser les questions suivantes :

- Quelles sont les données disponibles? Sous quelle forme se présentent-elles? En quelles unités sont-elles exprimées? Dans quel ordre sont-elles?
- Quelles sont les solutions adéquates? Sous quelle forme? En quelles unités doivent-elles être exprimées? Dans quel ordre doivent-elle être établies?
- Quelles méthodes sont utilisables pour résoudre ce problème? Laquelle est la plus efficace?

Les symboles les plus couramment utilisés, ainsi que leur définition sont indiqués à la figure 18. Les quelques exemples qui suivent illustrent la façon dont on utilise ces symboles dans les organigrammes.

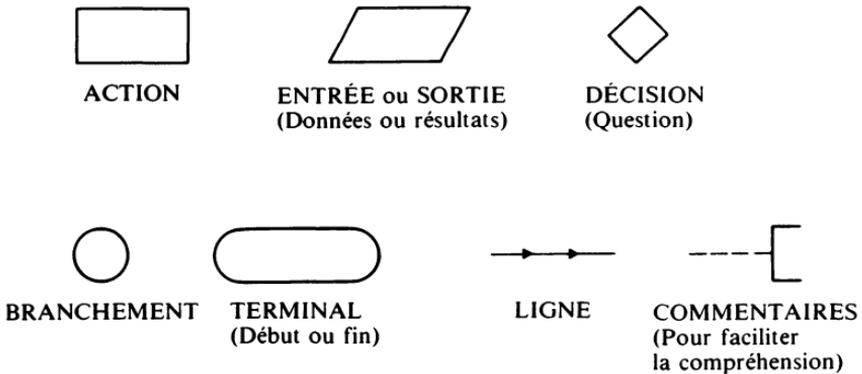


Fig. 18 Symboles de programmation.

Toutefois, avant d'écrire un organigramme, il faut supprimer certaines ambiguïtés possibles, en modifiant quelques caractères de la façon suivante :

- le chiffre zéro s'écrit \emptyset pour le distinguer de la lettre O,
- le chiffre un s'écrit 1 pour le distinguer de la lettre I,
- la lettre Z s'écrit \mathbb{Z} pour la distinguer du chiffre 2,
- la multiplication s'écrit (*) et l'exposant s'écrit (**), en fonction du langage utilisé.

EXEMPLE 1

Écrire l'organigramme permettant de téléphoner à un ami, à partir d'un téléphone public, en veillant à bien détacher chaque opération. La solution est donnée à la figure 19.

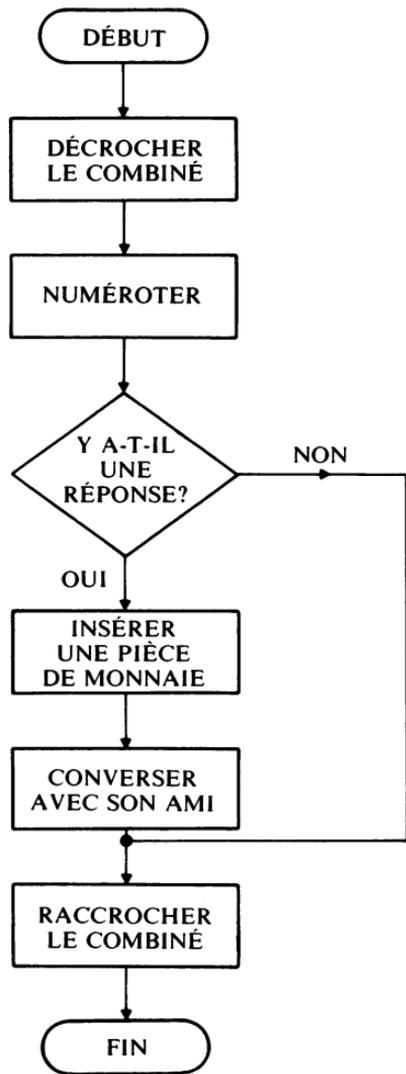


Fig. 19 Organigramme permettant de téléphoner à un ami.

EXEMPLE 2

Ecrire l'organigramme du réveil du matin. La figure 20 donne une des solutions possibles. Il est important de noter qu'il ne faut jamais rien laisser au hasard.

Symboles arithmétiques

Les deux exemples d'organigramme que l'on vient de décrire correspondent à des tâches quotidiennes. Il en est d'autres qui con-

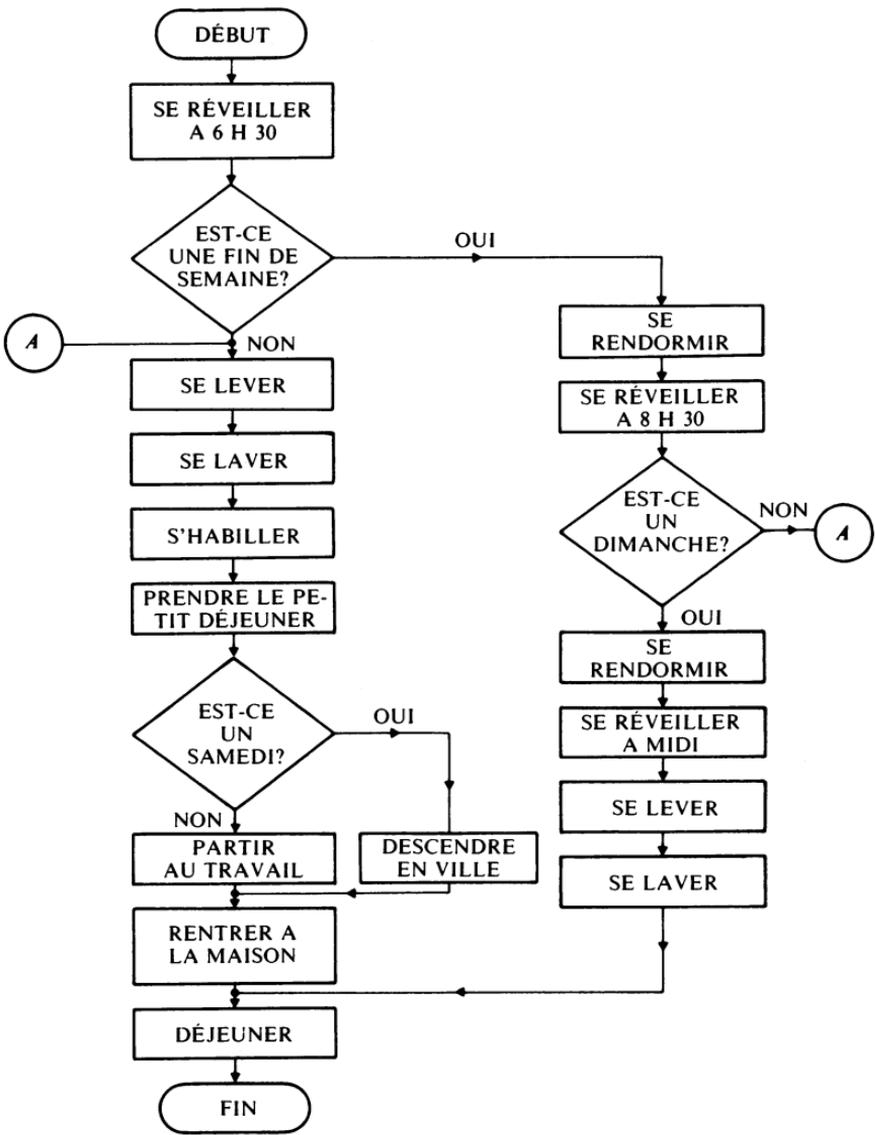


Fig. 20 Organigramme du réveil du matin.

cernent les opérations arithmétiques classiques. Dans ce cas, les organigrammes comportent des variables, qui comme en algèbre, s'expriment sous la forme d'une lettre ou d'un groupe de lettres, tel que x , y , a , b , A , N , NUM , $RÉP$, etc. Chaque variable occupe un emplacement dans la mémoire de l'ordinateur et sa valeur est spécifiée à tout moment par l'organigramme. L'attribution d'une valeur est généralement indiquée par l'un des trois signes suivants :

= : -

Par exemple, l'expression algébrique $P = Q$ indique que le contenu de la mémoire Q est attribué à la mémoire P , ce qui revient à dire que la variable P prend la valeur de la variable Q . L'expression algébrique $P \leftarrow Q$ revient à exprimer la même idée.

Expressions algébriques

Lorsque les opérations arithmétiques nécessitent l'utilisation de variables, on peut construire une expression algébrique en écrivant :

$$\text{Variable} = \text{Expression algébrique}$$

L'expression algébrique est composée d'une combinaison de variables et d'expressions arithmétiques classiques, tel que :

$$y = x + 1 \quad x = y + \frac{z}{a} \quad x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

La complexité de l'expression algébrique détermine généralement le langage de programmation à utiliser.

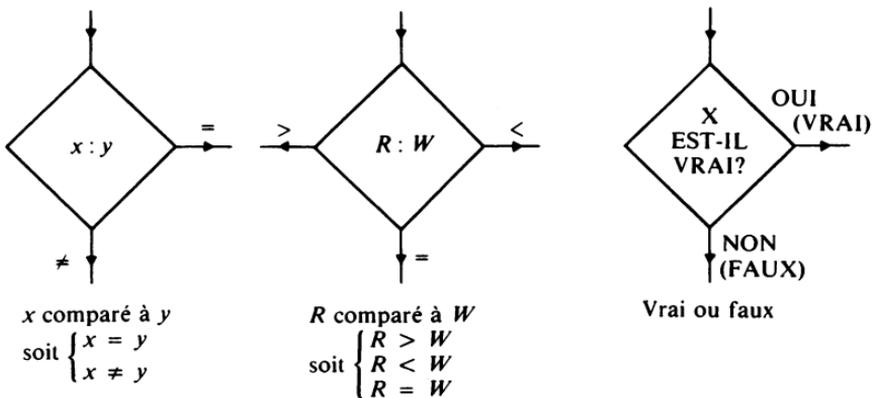


Fig. 21 Symboles de décision.

Les décisions se prennent par comparaisons, par opérations logiques ou par relations arithmétiques. Quelques variations autour du symbole de décision sont indiquées à la figure 21.

Bouclage

Le bouclage est un procédé permettant de répéter plusieurs fois une section d'un programme. Toutefois, lorsque l'on prépare un programme dans lequel on utilise la technique de bouclage, il

convient de s'assurer que l'on peut sortir de la boucle par l'un des deux moyens suivants :

- lorsque l'on sait à l'avance le nombre de fois que l'on doit parcourir la boucle. Dans ce cas, il suffit d'un compteur que l'on incrémente à chaque tour de boucle. La sortie de la boucle s'opère dès que le compteur a atteint la valeur prédéterminée.
- lorsque l'on sait à l'avance la valeur limite de la variable. Dans ce cas, il suffit de comparer à chaque tour de boucle la valeur de la variable avec sa valeur limite. Tant qu'elle n'est pas atteinte, on parcourt à nouveau la boucle. La sortie de la boucle s'opère dès que la variable a atteint ou dépassé la valeur limite.

EXEMPLE 3

Ecrire l'organigramme de calcul de Y à la puissance N . La figure 22 montre la solution trouvée, dans laquelle on utilise un compteur pour sortir de la boucle.

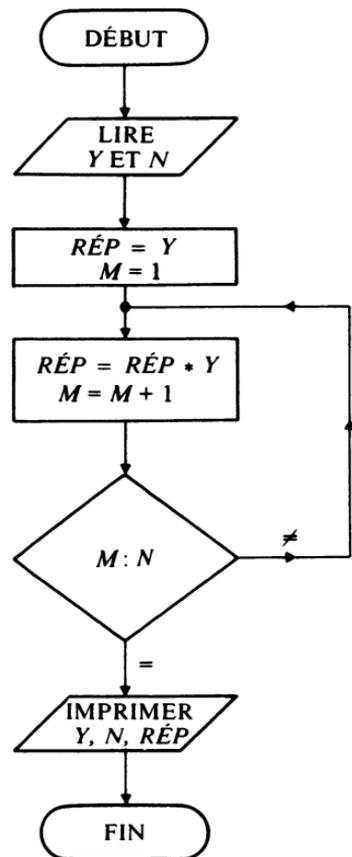


Fig. 22 Organigramme de l'exemple 3.

EXEMPLE 4

Écrire l'organigramme permettant de résoudre le problème suivant : on donne deux nombres positifs A et B devant former un troisième nombre C égal à la somme du carré du plus grand des deux avec l'autre. La figure 23 donne une solution à ce problème.

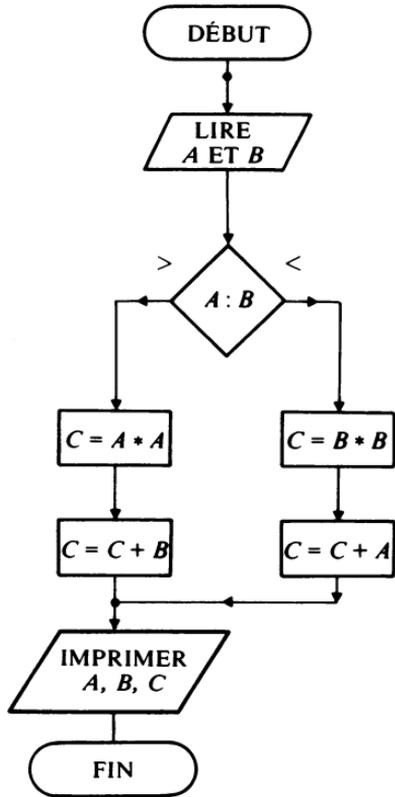


Fig. 23 Organigramme de l'exemple 4.

EXEMPLE 5

Écrire l'organigramme permettant de résoudre le problème suivant : on considère la variable A dépendant des paramètres P , Q , R , S , et T de la manière suivante :

- si $P = 2$, $A = P^2 + Q + R - S$
- si $P \neq 2$ et $Q = 3$, $A = T^2 + Q + R - S$
- si $P \neq 2$, $Q \neq 3$ et $R = 4$, $A = (P + T)^2 - Q - R + S$
- si $P \neq 2$, $Q \neq 3$ et $R \neq 4$, $A = + 3$

La solution à ce problème est donnée par l'organigramme de la figure 24, où l'on voit qu'une seule opération est exécutée à la fois.

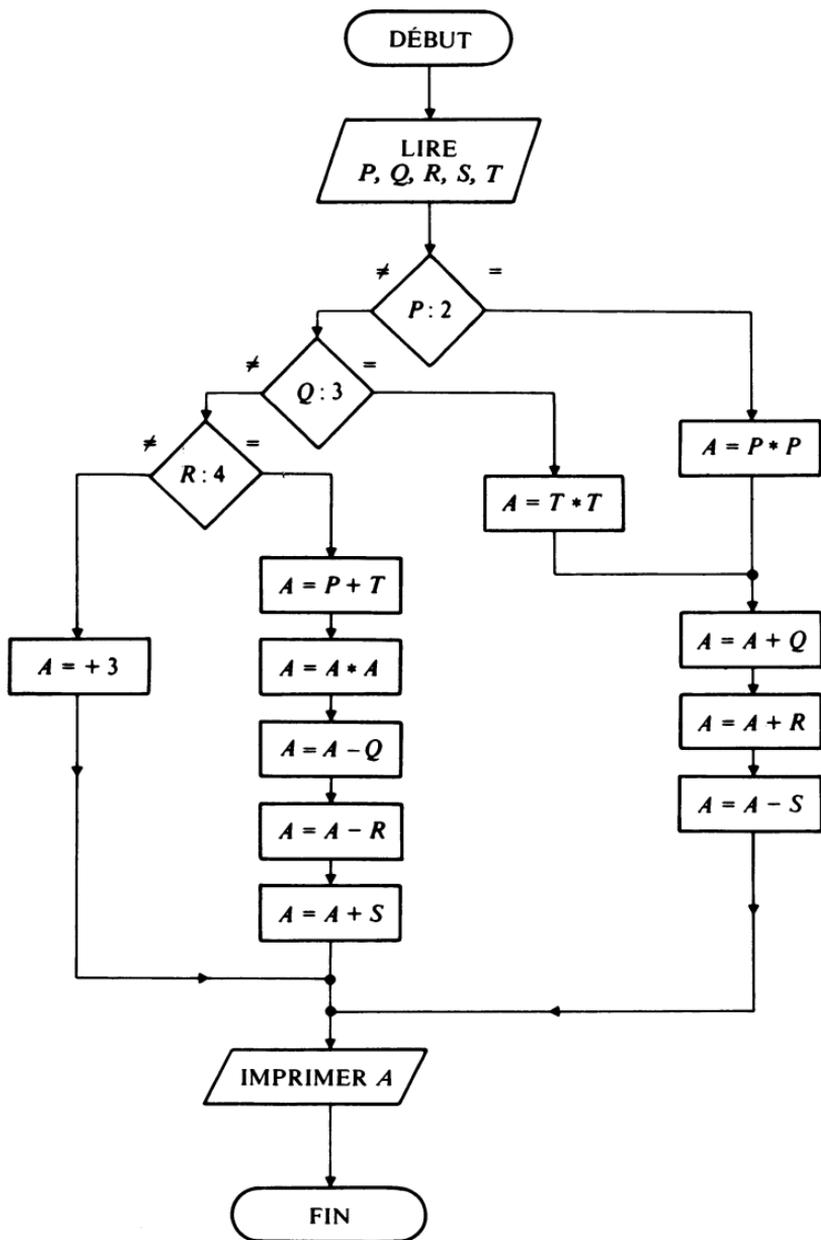


Fig. 24 Organigramme de l'exemple 5.

EXEMPLE 6

Écrire l'organigramme permettant de classer les différentes pièces de monnaie française. La solution est donnée à la figure 25.

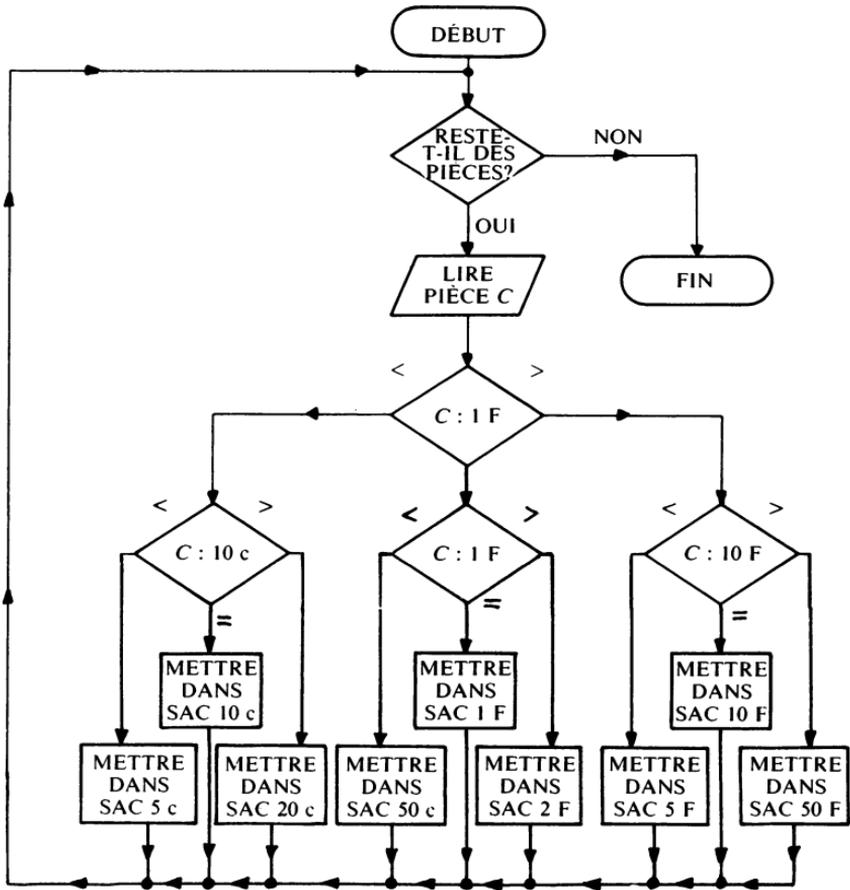


Fig. 25 Organigramme de l'exemple 6.

Le programme

Un programme comprend une suite d'instructions destinées à la machine. Ces instructions sont enregistrées dans des adresses consécutives de la mémoire de l'ordinateur. Le compteur ordinal explore ces instructions dans l'ordre, ce qui oblige l'ordinateur à les exécuter. Les opérations arithmétiques sont exécutées dans l'accumulateur, de même que les transferts de données.

Un exemple simple de programme de calcul de la paye est donné à la figure 26, dans lequel les données de calcul de la paye d'une personne sont inscrites sur deux cartes, la première indiquant le tarif

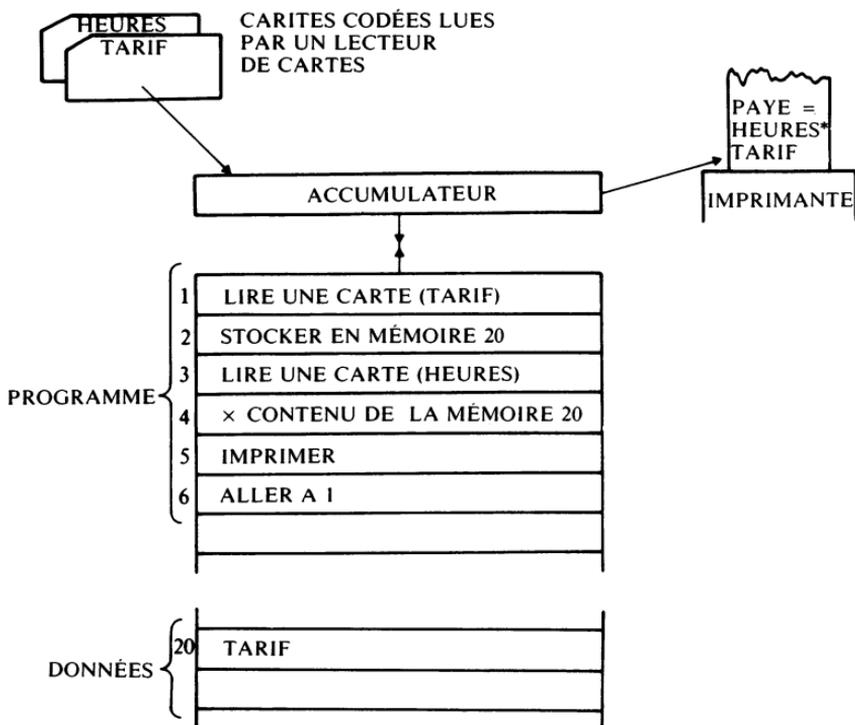


Fig. 26 Programme de calcul de la paye.

horaire, et la seconde indiquant le nombre d'heures de travail effectuées par cette personne. Il faut noter que les instructions précisent également le moyen d'entrée ou de sortie des données, tel que le lecteur de cartes pour les données d'entrée et l'imprimante pour les données de sortie.

Conception d'un logiciel

Dans un système informatique, quel qu'il soit, le logiciel fournit l'interface entre l'opérateur humain et la machine, se traduisant finalement en une suite d'instructions dont la forme est compréhensible par la machine. Il a déjà été question précédemment des difficultés que l'on peut rencontrer dans l'établissement des langages d'interface.

Divers niveaux de langage ont maintenant été mis au point entre la machine et le programmeur, que l'on peut classer en trois catégories :

- langage machine,
- langage assembleur,
- langages évolués.

Un programme écrit en langage machine comprend une liste d'instructions en binaire, qu'il est possible de charger directement dans la mémoire de l'ordinateur. Il est nécessaire de préciser en face du numéro de chaque adresse, si le mot binaire enregistré correspond à une instruction ou à des données.

On suppose une instruction typique du genre *additionner le contenu de la mémoire 50 (110010 en binaire) au contenu de l'accumulateur, en maintenant inchangé le contenu de la mémoire 50*. Dans ce cas, l'opération à effectuer est ADD, et l'adresse des données est 110010 (50 en décimal). En supposant que le code correspondant à l'opération d'addition soit 01, et que la longueur du mot soit de 8 bits, l'instruction se présente sous la forme :

Opérateur Adresse
 01 110010

EXEMPLE 7

Ecrire un programme en utilisant les instructions de machine de la figure 27, permettant de résoudre le problème :

$$Q = PU + \frac{QV}{R} - SW$$

Code opération n°	Instruction mnémonique	Instruction en clair	Opérations à effectuer
01	RAD	Remise à zéro et addition	Remise à zéro de la section arithmétique et addition du contenu de l'adresse _____ au contenu de l'accumulateur.
02	ADD	Addition	Addition du contenu de l'adresse _____ au contenu de l'accumulateur.
03	STO	Stockage	Stockage du contenu de l'accumulateur à l'adresse _____.
04	SOU	Soustraction	Soustraction du contenu de l'adresse _____ du contenu de l'accumulateur
05	MUL	Multiplication	Multiplication du contenu de l'accumulateur par le contenu de l'adresse _____.
06	DIV	Division	Division du contenu de l'accumulateur par le contenu de l'adresse _____.
07	IMP	Impression	Impression du contenu de l'accumulateur.
08	DÉBUT	Début de programme	Mise en marche de l'ordinateur et recherche de la première instruction.
09	FIN	Fin de programme	Arrêt de l'ordinateur.

Fig. 27 Code machine.

On suppose que les valeurs des variables sont dans les mémoires suivantes :

P est dans la mémoire 100
Q 101
R 102
S..... 103
U..... 200
V..... 201
W..... 202

On considère que le mot d'instruction fait 8 bits de longueur, ce qui donne le programme d'instructions en langage de machine de la figure 28.

Instruction mnémotique	Adresse de l'instruction	Instruction en binaire	Contenu de l'accumulateur
DÉBUT	000	08000001	0
RAD	001	01000200	U
MUL	002	05000100	P.U.
STO	003	03000300	P.U.
RAD	004	01000201	V
MUL	005	05000101	Q.V
DIV	006	06000102	Q.V/R
STO	007	03000301	Q.V/R
RAD	008	01000202	W
MUL	009	05000103	S.W
STO	010	03000302	S.W
RAD	011	01000300	P.U.
ADD	012	02000301	P.U+Q.V/R
SOU	013	04000302	P.U+Q.V/R-S.W
IMP	014	07000000	P.U+Q.V/R-S.W
FIN	015	09000000	

Fig. 28 Programme d'instructions en langage de machine.

La méthode qui consiste à écrire les programmes en langage machine est assurément laborieuse et difficile. De plus, la conception de ce type de programme prend trop de temps et les modifications sont délicates. Toutefois, le code de machine est bien adapté aux micro-ordinateurs utilisant un microprocesseur et des mémoires ROM, PROM ou EPROM pour stocker les instructions, car il permet de conserver à cet ensemble un coût relativement faible. Un assembleur est un programme spécial qui permet d'écrire les instructions sous la forme ADD 50 ou SOU 100 qui est automatiquement traduite en langage de machine. En général, une instruction mnémotique correspond à une instruction de machine. Ces programmes mnémotiques, tout en restant longs à écrire, sont plus pratiques à manipuler que les langages de machine.

Une liste d'instructions typiques pour un langage d'assembleur mnémonique est donné ci-après :

(A) indique le contenu de l'accumulateur,

(n) indique le contenu de la mémoire n,

(N) indique un nombre entier positif compris entre 0 et 999.

Instruction	Opération	Observations
CHA n	$(n) \rightarrow A$	(n) ne change pas
STA n	$(A) \rightarrow n$	(A) ne change pas
ADD n	$(A) + (n) \rightarrow A$	(n) ne change pas
SOU n	$(A) - (n) \rightarrow A$	(n) ne change pas
MLT n	$(A) * (n) \rightarrow A$	(n) ne change pas
DIV n	$(A) / (n) \rightarrow A$	(n) ne change pas
CHA N	$N \rightarrow A$	Cette suite d'instructions correspond au nombre entier position (N) et non à l'adresse (n) de la mémoire.
ADD N	$(A) + (N) \rightarrow A$	
SOU N	$(A) - (N) \rightarrow A$	
MLT N	$(A) * (N) \rightarrow A$	
DIV N	$(A) / (N) \rightarrow A$	

Fig. 29 Langage assembleur.

La figure 29 résume les différentes instructions du langage d'assembleur.

EXEMPLE 8

En utilisant les instructions de la figure 29, écrire le programme de l'opération :

$(x + 3) \cdot 40$

On suppose que x est contenu dans la mémoire 12 et que le résultat de l'opération est à inscrire dans la mémoire 12.

Solution :

CHA 12

ADDN 3

MLTN 40

STA 12

EXEMPLE 9

Ecrire le programme qui consiste à additionner deux nombres contenus dans les mémoires 25 et 26, à stocker le résultat de l'addition dans la mémoire 25 et à remettre à zéro la mémoire 26.

Solution :
CHA 25
ADD 26
STA 25
CHAN 0
STA 26

La plupart des langages assembleur possède un plus grand nombre d'instructions, instructions « saut », « modification » et « fonction », etc., permettant de réaliser des opérations plus complexes. Un langage assembleur couramment utilisé est le code ASCII (abréviation de l'expression anglaise : « American Standard Code for Information Interchange »).

Les langages évolués sont totalement indépendants de la machine, faciles à apprendre et permettent au programmeur de se concentrer sur les problèmes de programmation. Il existe deux types de langages évolués. Le langage interprétatif, tel le BASIC (abréviation de l'expression anglaise : « Beginners All-purpose Symbolic Instruction Code »), que l'on considère souvent comme un langage conversationnel, étant donné la forme des instructions. Ce type de langage est traduit en code machine par l'intermédiaire d'un *interpréteur*. Le deuxième type de langage évolué est converti en langage assembleur par l'intermédiaire d'un *compilateur*, avant la traduction finale en code machine. Le compilateur est un programme comprenant une suite d'expressions utilisées dans ce type de langage, et une liste d'instructions de machine à exécuter pour chaque expression. Ainsi, en employant un programme évolué d'entrée, de pair avec ce programme de compilateur, on obtient un programme de machine que l'on peut utiliser pour traiter les données. Pendant la compilation, le compilateur analyse chaque instruction du programme d'entrée, afin d'y déceler des erreurs de construction, et il donne les ordres nécessaires pour que ces erreurs soient imprimées. Les messages d'erreurs s'appellent des diagnostics, car ils aident grandement à « déterminer » un programme, c'est-à-dire trouver et corriger les fautes de ce programme. Pendant ce traitement, seules les erreurs non logiques seront détectées, puisque la machine ne peut détecter les erreurs logiques.

Plusieurs programmes évolués sont utilisés, et l'on peut citer entre autres :

- le FORTRAN (de l'anglais : « FORMula TRANslator »),
- le COBOL (de l'anglais : « COMmon Business Oriented Language »),
- l'ALGOL (de l'anglais : « ALGORithmic Language »),
- le CORAL (de l'anglais « Computer On-Line Real-time Application Language »). Ce langage est un développement de l'ALGOL.

- l'APL : (Advanced Programming Language).
- Le PASCAL.

Préparation d'un programme

On vient d'examiner assez rapidement les principes de programmation permettant de disséquer complètement un problème et d'écrire le programme correspondant. Lorsque la tâche que l'on demande à la machine d'exécuter est simple, on peut écrire directement le programme en langage machine, instruction après instruction, en manipulant les commandes du panneau avant de l'ordinateur. Quoique cette méthode soit la plus utile dans le cas d'utilisation de micro-ordinateurs, elle devient trop laborieuse dès qu'il s'agit d'un programme plus élaboré.

On écrit donc les programmes plus élaborés sur des feuilles de programmation qui servent ensuite à perforer des bandes ou des cartes perforées que l'on donne à lire à l'équipement périphérique correspondant. On peut également, taper directement le programme sur une télétype ou sur une console de visualisation.

La console de visualisation est l'équipement terminal le plus performant, car elle permet à l'opérateur d'écrire le programme grâce au clavier, et de recevoir des retours d'informations de la machine grâce à un écran de visualisation, ressemblant à un écran de télévision. Cet équipement est de loin celui qui autorise la plus grande facilité et rapidité de dialogue avec la machine.

et pour conclure

quelques compléments

Symboles logiques

Diverses formes de symboles ont été, ou sont utilisés à travers le monde. La symbolique la plus répandue est la symbolique américaine que l'on peut voir à la figure A.1 *b*.

Toutefois, depuis quelques années, une symbolique européenne est en train d'apparaître. On voit d'après la figure A.1 *a* qu'elle est plus facile à dessiner que la symbolique américaine, car elle est constituée de rectangles. La plupart des industriels français empruntent désormais cette symbolique européenne.

Brochage des circuits de la série TTL 74

L'ensemble des figures A.6 donne le brochage des circuits intégrés dans cet ouvrage. Le brochage des circuits est toujours indiqué en vue de dessus, la petite encoche étant à gauche. La désignation des circuits a également subi une standardisation, car tous les fabricants ont sensiblement les mêmes produits à leur catalogue. C'est ainsi que, généralement, les trois derniers chiffres de désignation sont les mêmes pour un type de circuit donné.

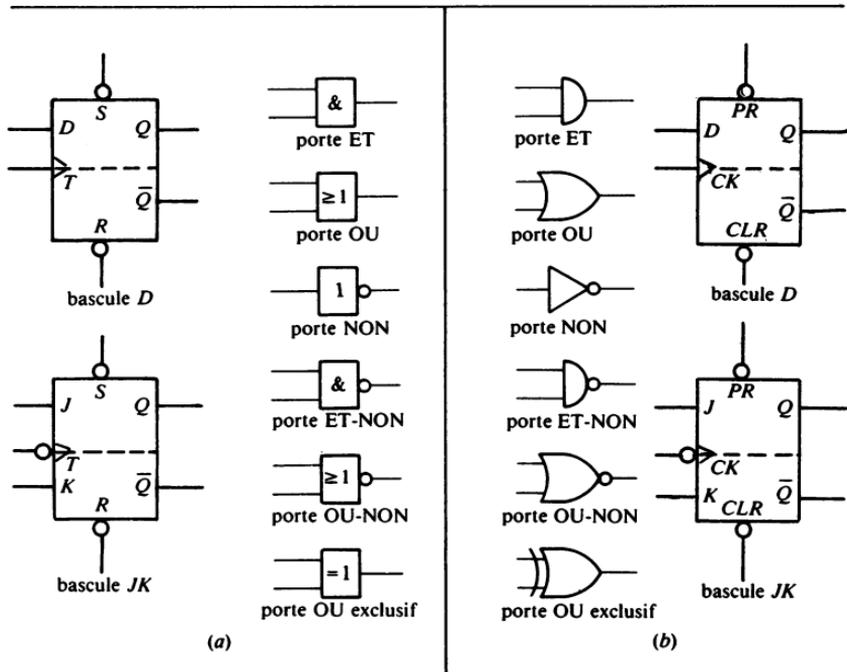


Fig. A.1 Symboles logiques :

(a) symboles européens actuels,

(b) symboles américains.

Quelques-unes des abréviations les plus courantes figurent dans la liste ci-après :

- NR** non raccordé
- Et** entrée
- St** sortie
- T** horloge
- S** forçage à « 1 »
- R** forçage à « 0 »
- RAZ** remise à zéro
- EFF** effacement
- VAL** validation

Quelques derniers conseils

Lorsque l'on utilise des portes logiques, il ne faut jamais laisser une entrée en l'air. Les entrées inutilisées doivent être raccordées au « 1 » logique ou au « 0 » logique, suivant qu'il s'agit de porte ET et ET-NON, ou de porte OU et OU-NON. Il est également possible de raccorder ensemble deux entrées lorsque l'une des deux ne sert pas. Par contre, il ne faut jamais raccorder ensemble

deux sorties de portes TTL classiques. Seules les portes à collecteur ouvert sont prévues pour cela.

La plupart des circuits MSI sont validés par l'application d'un « 1 » logique. Toutefois, les entrées *S* et *R* de forçage des bascules ne sont actives que si l'on applique un « 0 » logique. La figure A.1 le montre très clairement, car les entrées en question sont précédées d'un rond indiquant l'inversion.

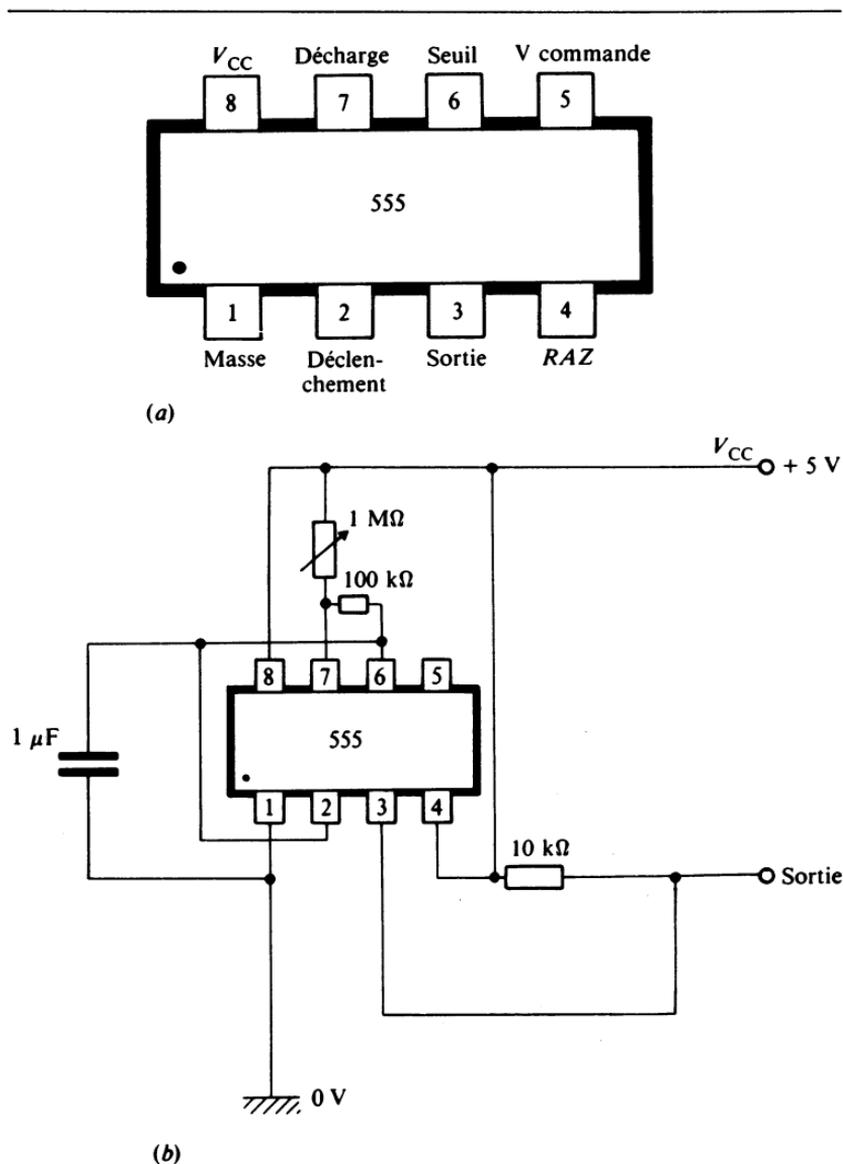


Fig. A.2. Générateur d'horloge 555 :

- (a) brochage,
- (b) schéma de principe.

Les entrées d'horloge sont toujours indiquées par une flèche (fig. A.1). Lorsque la flèche n'est pas précédée d'un rond, la bascule change d'état au front montant de l'impulsion d'horloge (bascule *D*). Lorsque la flèche est précédée d'un rond, la bascule change d'état au front descendant de l'impulsion d'horloge (bascule *JK*).

Générateur d'horloge 555 et circuits divers

• Le générateur d'horloge 555.

Ce circuit permet de réaliser un générateur d'horloge à haute stabilité. Le brochage de la version DIL est fourni à la figure A.2 a. Le schéma de la figure A.2 b est un générateur d'horloge dont la fréquence peut varier entre 1 Hz et 10 Hz, grâce à un potentiomètre de 1 M Ω . Il est possible de changer de gamme de fréquences, en modifiant la valeur du condensateur *C*.

• Circuit antirebonds

Lorsque l'on utilise des interrupteurs ou des boutons poussoirs mécaniques pour fournir des signaux logiques d'entrée, le rebondissement des contacts peut causer des problèmes. En effet, les signaux logiques peuvent être affectés de variations rapides entre les niveaux logiques « 0 » et « 1 », lors du basculement d'un interrupteur. Pour éliminer cet inconvénient, on place un inver-

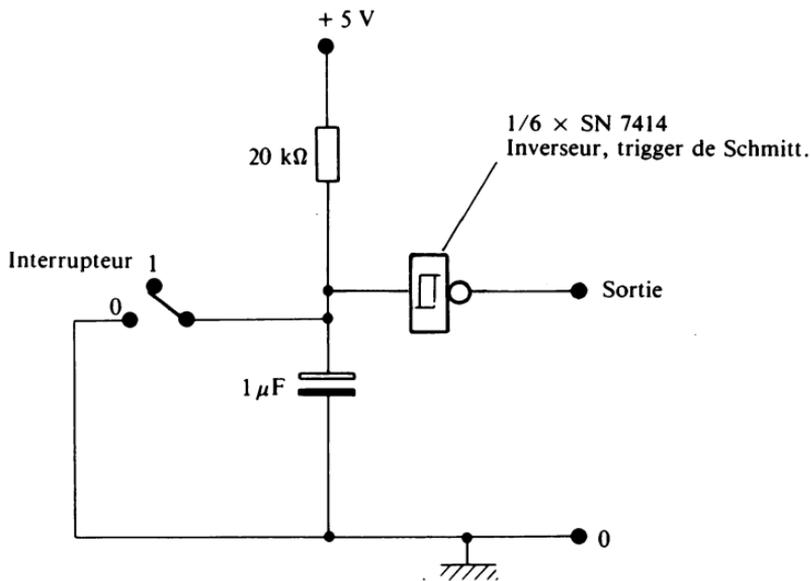


Fig. A.3. Circuit antirebonds.

seur *trigger de Schmitt* derrière un interrupteur, tel que le montre la figure A.3. Cet inverseur « nettoie » les fronts de commutation de l'interrupteur.

- *Multivibrateur astable.*

Une méthode économique pour fabriquer un générateur d'horloge, consiste à utiliser deux portes ET-NON (NAND) que l'on branche comme indiqué à la figure A.4. Les valeurs typiques des deux résistances et des deux condensateurs sont comprises entre 470Ω et $10 \text{ k}\Omega$ et entre $0,22 \mu\text{F}$ et $1\,000 \mu\text{F}$ suivant la fréquence d'oscillation.

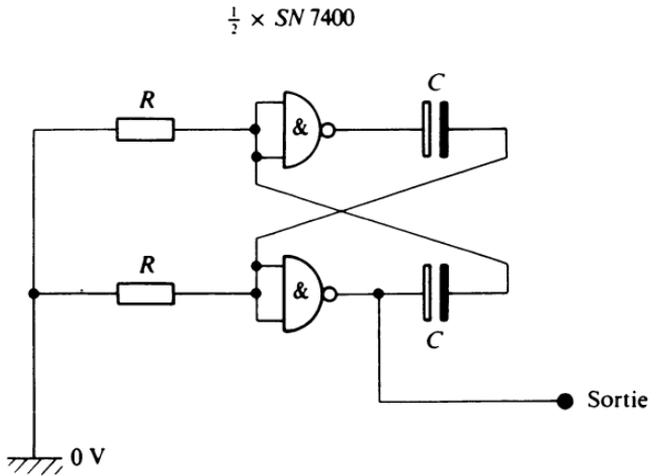


Fig. A.4. *Multivibrateur astable.*

Et pour terminer :

- *Le dé électronique.*

Il existe plusieurs manières de réaliser un dé électronique. Celle de la figure A.5 utilise le multivibrateur astable décrit précédemment. Ce multivibrateur oscillant à une fréquence voisine de 1 kHz , alimente une décade SN 7490 câblée en compteur modulo 7, lorsque l'on appuie sur le bouton « appuyer pour secouer le dé ». Il fournit donc à ce compteur un nombre inconnu d'impulsions, et le résultat compris entre 1 et 6 s'affiche sur l'afficheur sept segments. La broche *Et EFF* est mise au « 0 » logique, ce qui empêche l'affichage du chiffre 0.

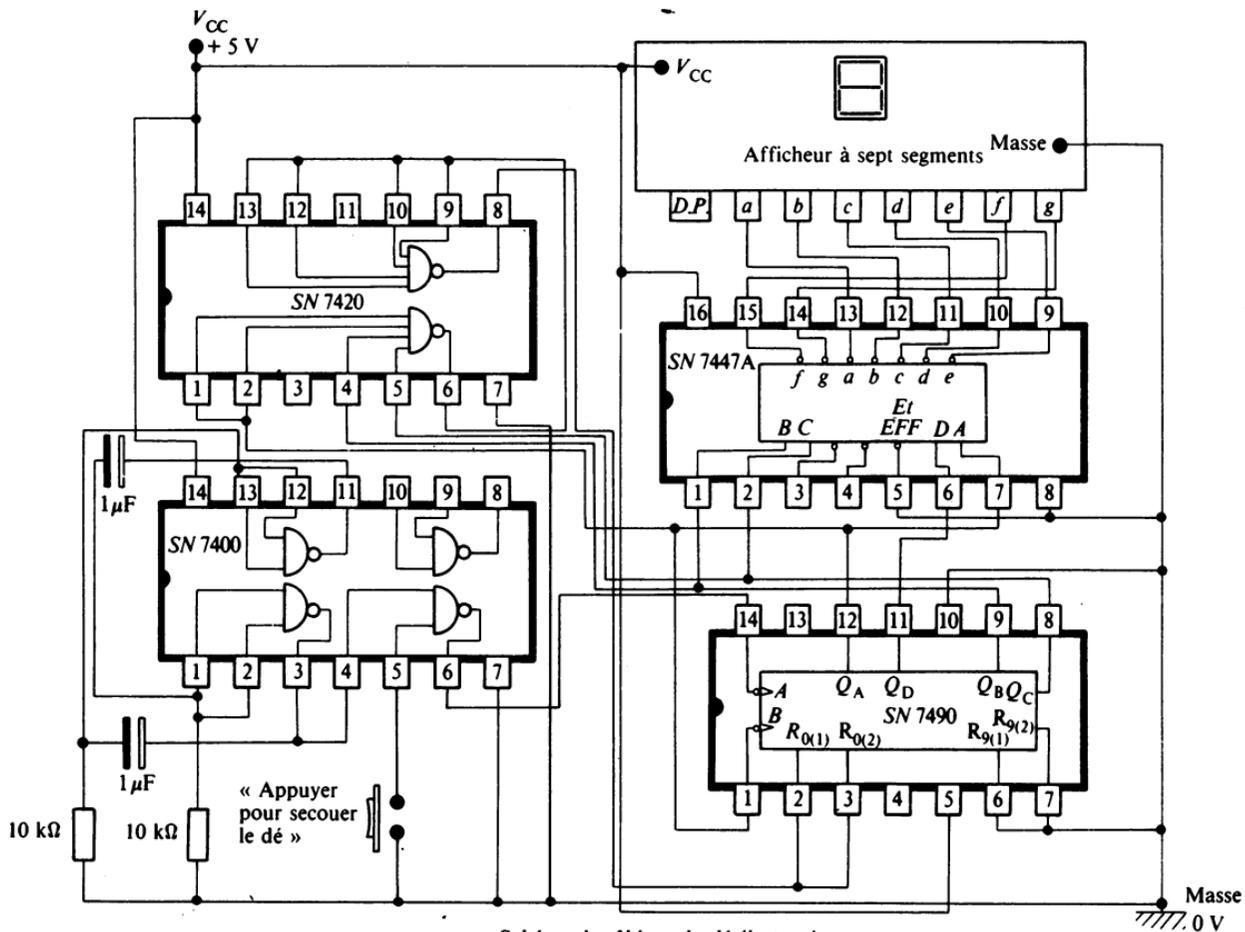


Schéma de câblage du dé électronique.

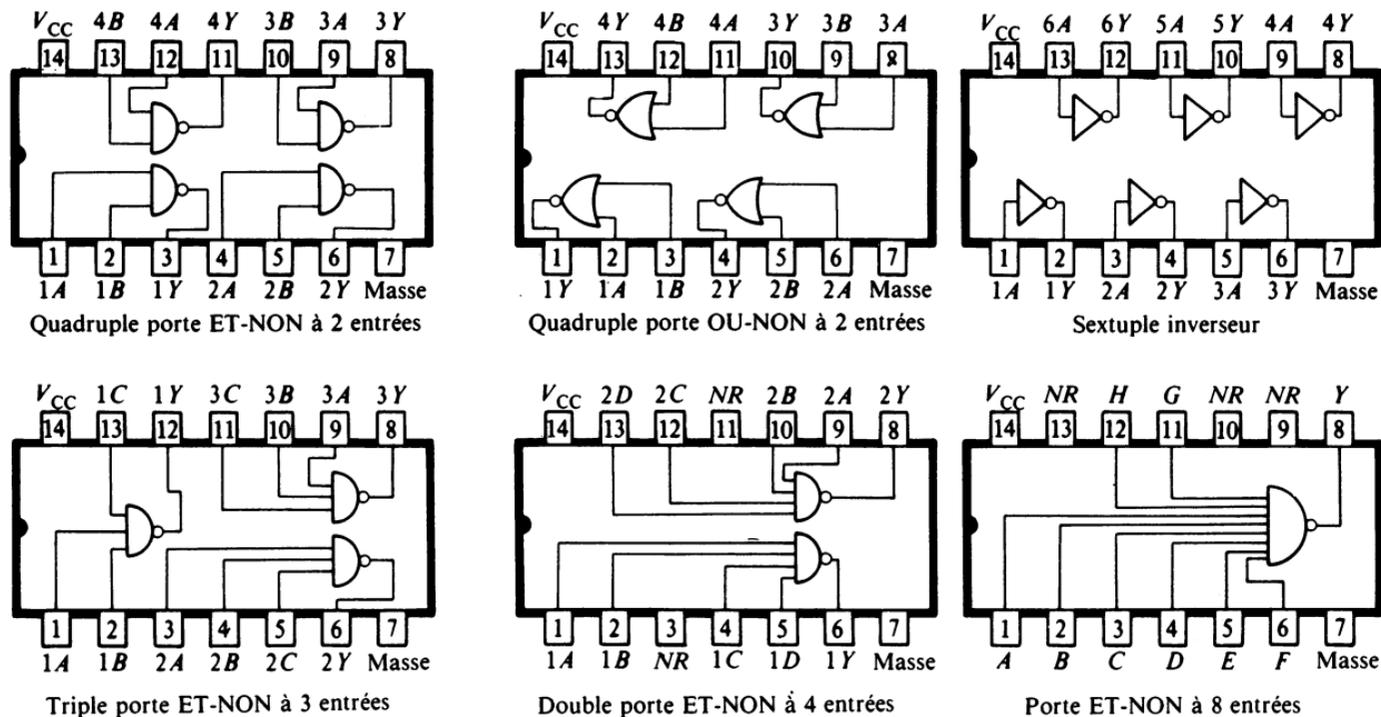
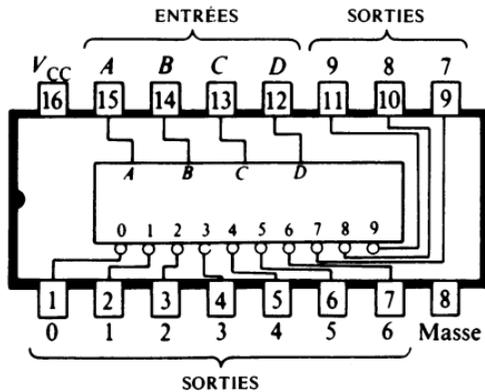
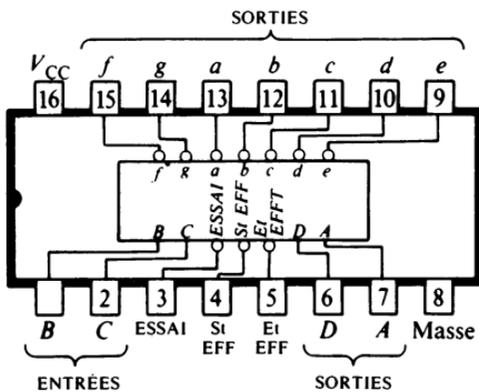


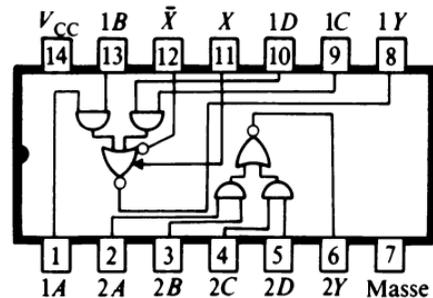
Fig. A.6 Brochage des circuits de la série TTL 74.



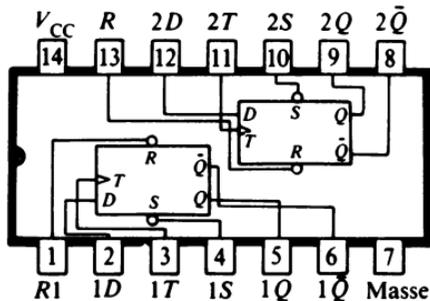
Décodeur DCB-décimal



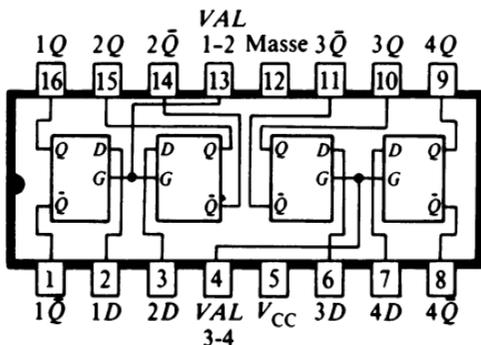
Décodeur sept segments



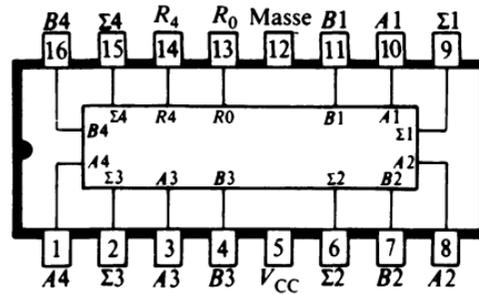
Double porte ET-OU-NON à 2 entrées



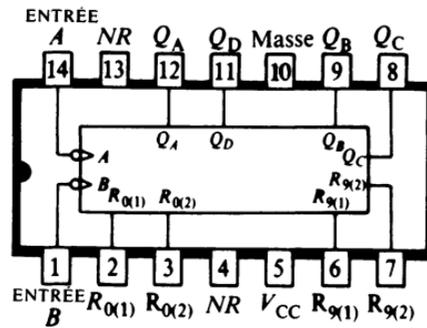
Double bascule D



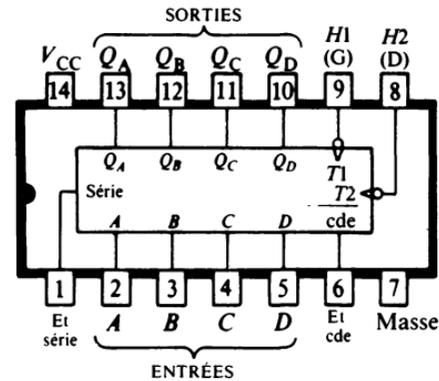
Quadruple bascule DT recopie



Additionneur complet à 4 bits.

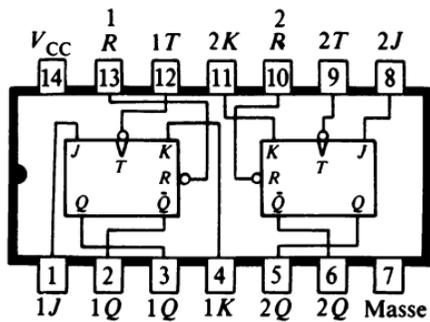


Décade asynchrone

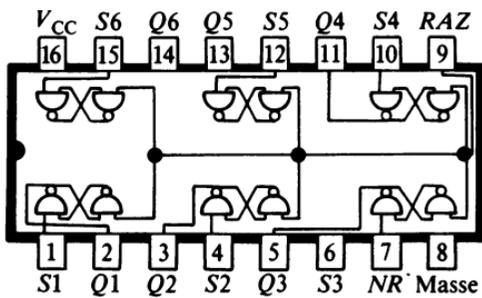


Registre à décalage à 4 bits

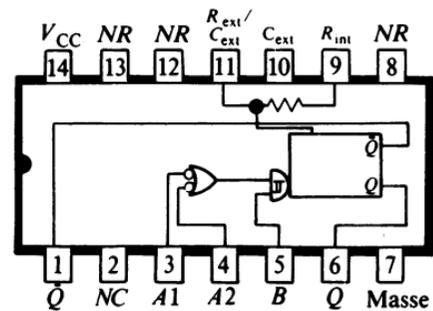
Fig. A.6. Brochage des circuits de la série TTL 74 (suite).



Double bascule JK



Quadruple bascule RS



Monostable

index

- abréviations, 13.
- accumulateur, 190.
- addition, 21.
 - binaire, 81.
- additionneur complet, 81, 199.
 - parallèle, 81.
 - série, 81.
 - soustracteur, 90.
- adressable, 128.
- adresse, 155.
- afficheur à LED, 103.
 - alphanumérique, 21.
 - à sept segments, 115.
- algèbre de Boole, 15.
- algorithme, 180.
- alphanumérique, 21.
- ALU (unité arithmétique et logique), 154.
- amplificateur de régénération, 169.
 - symétrique, 38.
- anticoïncidence, 53.
- antirebonds, 202.
- appareil de mesure, 1.
- ASCII, 193.
- assembleur, 191.
- bande magnétique, 177.
- barrière de potentiel, 26.
- bascule D, 76, 80.
 - JK, 78, 80.
 - RS, 73.
 - RST, 76.
- base, 3, 21.
- BASIC, 193.
- BCD, 117.
- binaire, 21.
- bipolaire, 8.
- bit, 70.
- bloc de données, 178.
- blocage, 37.
- bouclage, 184.
- bulle magnétique, 160.
- cabestan, 177.
- calculateur, 151.
- canal, 8.
- capacité de collecteur, 5.
 - parasite, 27.
- caractère, 177.

- caractéristique de commutation, 26.
 - de transfert, 28.
 - de sortie, 10.
 - électrique, 43.
- cascade, 33.
- case adjacente, 61.
 - optionnelle, 71.
- CCD (dispositif à couplage de charge), 160.
- champ électrique, 8,
 - magnétique, 165.
- charge électrique, 26.
 - standard, 43.
- chute de tension, 33.
- circuit intégré, numérique, 1.
 - - linéaire, 1.
 - - monolithique, 7.
- claquage, 174.
- CMOS, 11.
- code d'opération, 155.
 - BCD, 117.
- collecteur commun, 44.
 - ouvert, 201.
- commande d'écriture, 100.
 - numérique, 26.
- comparateur, 53.
- compilateur, 193.
- complément à deux, 87.
- compteur asynchrone, 108.
 - en anneau, 95.
 - Johnson, 95.
 - ordinal, 156.
 - synchrone, 14.
 - décompteur, 110.
- convertisseur parallèle-série, 107.
- courant d'évacuation, 40.
 - de drain, 9.
 - de fuite, 26.
 - de magnétisation, 162.
 - de source, 40.
 - direct, 25.
 - inverse, 25.
- CPU (unité centrale de traitement), 153.
- cycle de comptage, 131.
- débordement, 87.
- décade, 114-199.
- décalage en tension, 33.
- décimal codé binaire (BCD), 117.
- décodeur BCD-décimal, 198.
 - sept segments, 198.
- demi-additionneur, 81.
- diagramme de Veitch, 60.
 - des temps, 109.
- diaphonie, 27.
- DIL (à double rangée de broches), 11.
- diode, 7.
 - au germanium, 30.
 - au silicium, 30.
 - de décalage, 33.
 - électroluminescente (LED), 40.
 - Schottky, 26.
- disque magnétique, 178.
- diviseur par deux, 79.
- EAROM, 171.
- ECL, 29.
- entrance, 25.
- entrée 48, 76, 101, 102.
- entrefer, 176.
- EPROM, 171.
- équation booléenne, 18.
- esclave, 78.
- fonctions ET, ET-NON, 46, 49.
- fonction NON, 49.
 - s OU, OU-NON, OU exclusif, 44, 53.
- format (d'adresse), 155.
- générateur d'horloge, 14.
 - pseudo-aléatoire, 127.
- grille isolée, 11.
 - métallique, 11.
- hexadécimal, 21.
- immunité (aux bruits), 24.
- impédance d'entrée, 10.
- inhibition, 164.
- instruction, 155.
 - mnémonique, 191.
- inverseur, 18.

jonctions, 5, 10, 26, 38.

langages, 152, 189.

lecteur de bande, 157.
– de cartes, 157.

logigramme, 49.

logique (différents types de) 14, 16,
29, 39, 160.

LSI, 45.

marge de bruit, 27.

mémoires (différents types de),
154, 155, 157, 160, 165, 166, 171,
173, 175, 177, 178.

microordinateur, 160.

multiplicateur d'impulsions, 94.
– parallèle, 92.
– binaire, 91.

octal, 21.

ordinateurs, 151, 158, 180.

parité, 162.

périphérique, 154.

portes (de base), 18, 19, 24, 40, 53.

programme, 152, 191.
PROM, 171.

RAM, 155.

registres, 95, 104, 156, 163, 199.

ROM, 155.

séquence, 100.

seuil d'entrée, 27.

sortance, 25.

sortie parallèle, 101.
– série, 102.

symboles, 195, 196.

systèmes (de numération), 21.

tables (de vérité), 18.

tableau (de Karnaugh), 59.

TEC, 8.

temps d'accès, 165.

têtes (diverses), 175, 176.

transition, 26.

transistors (divers types), 2, 3, 6, 8,
9, 11, 35, 38, 45, 46.

unité arithmétique et logique
(ALU), 154.

1-3

conquérir la logique, B. Woollard

PRATISERVICE LECTEURS

Nom :

Prénom :

Age :

Profession :

Adresse :

.....

Activité principale de loisir (ou centre d'intérêt)

.....

souhaite être informé des nouveautés paraissant
dans la collection :

oui - non

dans une autre collection

oui - non

Si *oui*, laquelle

Sujet

Commentaires sur le présent ouvrage :

.....

.....

Suggestions : (améliorations, sujets traités...)

.....

.....

Comment avez-vous connu cet ouvrage?

librairie - publicité - salon - autre

lieu d'achat :

date : prix :

Chez le même éditeur

MODULES TECCART

Destinées à tous ceux qui souhaitent développer leurs connaissances sur les semi-conducteurs et le fonctionnement des systèmes de communication, deux séries résumant d'une manière claire et très didactique les cours techniques de l'Institut Teccart de Montréal :

- Série **COMMUNICATIONS** par D. BENSOUSSAN

Les antennes, 160 p.

La modulation, 112 p.

Émetteurs et récepteurs, 112 p.

Reproduire le son, 144 p.

- Série **SEMI-CONDUCTEURS** par V. RAMIREZ

Physique des semi-conducteurs, 112 p.

Les diodes modernes, 176 p.

*Les résistances non linéaires
à semi-conducteurs, 144 p.*

PRATIGUIDE

électronique

Conquérir la logique

B. WOOLLARD

Mystérieuses boîtes noires!

Rien de plus désagréable qu'un montage à circuits logiques qu'une impulsion parasite fait fonctionner de manière illogique.

On peut éviter cela en apprenant à bien connaître les comportements des circuits logiques fondamentaux et les précautions d'emploi. Cet ouvrage, par de multiples montages pratiques, fournit les bases qui permettent d'aborder microprocesseurs et microordinateurs.



ISBN 2-04-010665-0
RETRONIK.FR 2023